High Level Systems Architecture

Data Access and Dissemination System (DADS)

Version No: 1.2

Version Publishing Date: July 11, 2005

Author: DADS Project Team

Owner: DADS Architecture Team

Client: Bureau of Census (BOC)

Contract Number: 50-YABC-7-66012

DOCUMENT ADMINISTRATION

Ensure that this document is current. Printed documents and locally copied files may become obsolete due to changes to the master document.

The document original is located in the project's documentation repository at the following file address:

I:\DADS Architecture\System Documentation\08 Wprod & Deliv\High Level System Architecture\HL System Architecture v1.2.doc

i. Revision History Log

The maintenance of this Revision History Log is mandatory. Document users discovering that the last logged revision date is more than 120 days old should assume the content is dated and should alert the document's owner.

Revision Number	Revision Date	Summary of Changes	Team - Author
V1.0	09.30.04	First baseline	DL, TN, MC, IM
V1.1	06.30.05	Second baseline; incorporated feedback from reviews, and well as corrections to AQS system; new DADS Context Diagram, and updates to reflect software updates to AFF to WebSphere, ESRI, development environment.	IM
V1.2	07.11.06	Third baseline; moved AFF, AQ, and DPP diagrams and tables with explicit IP addresses to a separate attachment	IM

ii. Identification

This document is identified as the Data Access and Dissemination System (DADS) High-Level System Architecture. The production and maintenance of this document is the responsibility of the DADS Architecture Team.

iii. Document References

The following documents should be reviewed as an extension of this document's subject matter or as supplemental reference material.

- American FactFinder Technical Design
- Advanced Query Technical Design
- Data Product Production Technical Design

TABLE OF CONTENTS

TABLES

FIGURES

1. INTRODUCTION

1.1. Scope

This document provides a high-level overview of the Data Access and Dissemination System (DADS) system architecture and describes:

- System architecture design influences
- Functional capabilities from an architectural perspective
- High-level operational scenarios
- System software
- Hardware architecture
- Network architecture
- The functional roles of system software
- The current physical and logical deployment of system software
- The current physical and logical deployment of hardware and network resources

1.2. Audience

This document is targeted at management and technical personnel who need a high-level introduction to DADS and to DADS' place within the BOC.

1.3. Purpose

This document is intended to answer the following questions for management and technical personnel needing a high-level introduction to DADS:

- What is DADS?
- What are the major DADS components?
- Who are the users of DADS?
- What provides the data sources used by DADS?
- Where is DADS installed?
- What design influences guided the current DADS architecture?
- How does DADS do what it does?

The above questions are answered from the following perspectives:

1. Functional view
2. Logical view
3. Physical view
4. Operational view

2. DADS FUNCTIONAL CAPABILITIES

The Data Access and Dissemination System Office (DADSO) provides tabulation and dissemination services to a broad range of stakeholders both inside and outside the U.S. Census Bureau. To avoid confusion, this document defines the following naming conventions:

- The DADSO organization is also referred to this as *"DADSO"* or simply *"the Office"* in this document. This term is useful when we need to highlight DADSO as organizational entity with the U.S. Census Bureau.
- The set of systems developed by DADSO are called the "*DADS systems."* Generally, the DADS systems refer to the set of IT resources and systems owned, managed, developed by the Office.
- The "*key DADS systems"* refers collectively to the trio of systems American FactFinder (AFF), Advanced Query System (AQS), and Data Product Production (DPP)

DADSO provides tabulation and dissemination services through its three primary systems – American FactFinder (AFF), Advanced Query System (AQS), and Data Product Production (DPP). Each system was developed at approximately the same time by different development teams using different tools, design approaches, and platforms. In hindsight, these systems have many common elements (e.g., processing geographic files from Geography Division, tabulating microdata, building tools for the *Reviewer* actor, etc.) that could have been developed centrally and shared across teams. In theory, the resulting DADS systems could have been developed faster and with more code reuse. In practice, this common design and reuse did not happen. AFF, DPP, and AQS had very different business requirements, nonfunctional requirements, disclosure constraints, and delivery schedules that made such reuse impractical.

2.1. System Context

The System Context represents the entire DADS system as a single object and identifies the interfaces between the system and external entities. The System Context Diagram defines the DADS system and identifies the information and control flows that cross the system boundary.

The System Context highlights several important characteristics of the system: users, external systems, and batch inputs and outputs. The objects within the system boundary define the scope of the DADS system and represent areas over which DADSO has control. The users and systems outside the boundary of the system are those that affect the system operation and development but are beyond the control of DADSO.

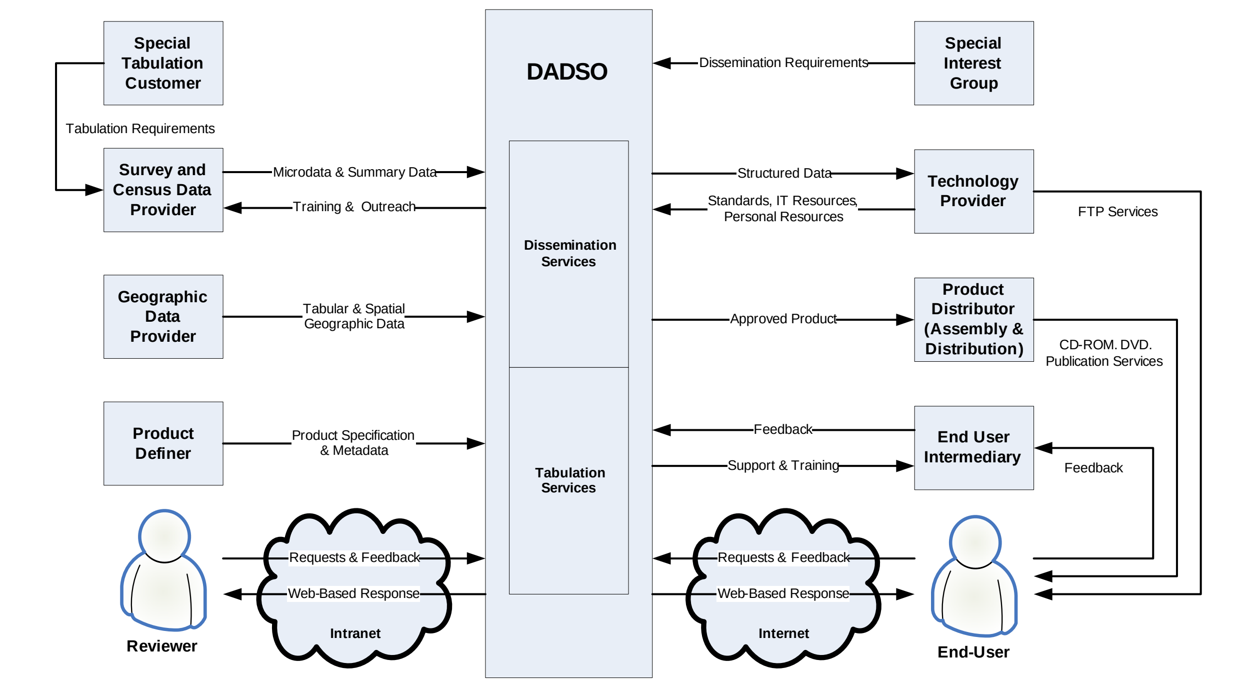

Figure 1: System Context

2.1.1. Actors and External Systems Interacting with DADS

2.1.1.1. Survey and Census Data Provider

The *Survey and Census Data Providers* are any one of a number of Census directorates that provide inputs of either microdata or summary data, and the associated metadata. These inputs, along with inputs from the *Geographic Data Provider* and the *Product Definer,* are used in the tabulation and dissemination process.

The following table provides a summary of the *Survey and Census Data Providers* supported by DADSO. Note that some programs disseminate data on separately maintained web sites. For example, the Demographic Directorate Population Estimate program manages their own dissemination services – only a subset of their data is disseminated on the DADS systems.

Directorate	Program	Years Supported	Tabulation Services	Dissemination Services
Decennial	Decennial Census	1990 – United States	No	Yes
		2000 – United States and Puerto Rico	Yes	Yes
		2000 – Island Areas	No	Yes
		2000 – Puerto Rico (Spanish)	No	Yes
	American Community Survey	1996 forward – United States	No	Yes
Demographic	Population Estimates	2001 forward – United States and Puerto Rico	No	Yes (partial)
	School Districts Data File	2000 – United States	Yes	No
Economic	Economic Census	1997 – United States	No	Yes (partial)
		2002 – United States	No	Yes
		1997 – Puerto Rico and Island Areas	No	Yes
	Non-Employer Statistics	2002 – United States	No	Yes

	Survey of Business Owners	2002 – United States	No	Yes
	Business Expenditure Survey	2002 – United States	No	Yes
	American Survey of Manufactures		No	Yes

Table 1: Summary of DADS Survey and Census Data Providers

2.1.1.2. Geographic Data Provider

The *Geographic Data Provider* provides both tabular and spatial geographic data. Spatial data are used for map products provided by AFF. Tabular data are used in both the tabulation and dissemination process.

2.1.1.3. Product Definer

The *Product Definer* represents the subject matter experts who provide detailed specifications for the content, structure, presentation and dissemination method for each product that is tabulated or disseminated.

2.1.1.4. Reviewer

The *Reviewer* verifies that products are built according to specification and reviews and clears them for public release. The review is conducted on internal Intranet systems in a secure environment by the same subject matter experts who define the products.

2.1.1.5. Product Distributor

The *Product Distributor* assembles the approved products for distribution through various media: Internet, CD-ROM and DVD, FTP and Abode Acrobat (PDF).

DADSO is primarily responsible for the Internet distribution of products and data, but leverages the services of the *Product Distributor* for other outputs. For example, the Administrative and Customer Services Division (ACSD) receives the following outputs from DADSO:

- Decennial summary file data for CD/DVD distribution and sale to the public
- Image-ready Population and Housing Characteristics (PHC) PDF files for assembly into publication quality PDF files for printing and dissemination

2.1.1.6. End User Intermediary

The *End User Intermediary* is the first line of contact for end users. These intermediates may be call centers, regional offices, and support staff for a wide network of secondary distributors of Census Bureau data and information. DADS provides training and technical support to each, who in turn, interact with end users of the data. The central point of contact for this extended community is the Marketing Services Office (MSO).

2.1.1.7. Technology Provider

The *Technology Provider* defines information technology standards and provides personnel and infrastructure resources to support automated dissemination. Services provided include:

- Support for public FTP access to resource managed by DADS, such as Decennial Summary Files and Economic Census 2002 data and metadata.
- Network infrastructure services, including security scans, DMZ support, firewalls, etc.
- COTS Concord performance monitoring services for all DADS UNIX and Windows servers
- Definition of IT standards for such areas as Internet, Security, and Web Accessibility guidelines.
- Intranet web site support
- Web site usability testing services

2.1.1.8. End-User

End-Users are the public customers of the Internet-based systems developed by DADSO. Some DADS systems, like Beyond 20/20 and the Advanced Query System, require user authentication; others like American FactFinder are open the public and require no login or password. AFF users are further classified into role-based categories – Surfers, Portrayers, Extractors, and Manipulators – based on how they used the services provided by AFF. *End-Users* may also seek on-line assistance by submitting feedback.

2.1.1.9. Special Interest Group

Special Interest Groups seek to serve their constituency by providing a customized view of Census data. For the Congressional Affairs Office (CAO) and the Race and Ethnic Advisory Committee (REAC), DADS has a separate user interface targeted to meet their needs. These groups generally repurpose existing data (as opposed to supplying new data) to create a customized view that meets the needs of their constituency.

2.1.1.10. Special Tabulation Customer

As a tabulation service provider, DADSO occasionally fulfills special tabulation requests from organizations external to Census. Generally, DADSO works with these *special tabulation customers* through an authorized Census organization like POP or HHES, who handle the legal and policy issues related to the special tabulation request.

For example, the DPP system produced a series of School District tabulations using the SF3 data, and a set of additional supplemental recodes, for the National Center for Education Statistics (NCES), but POP and HHES played the roles of *Product Definer*, *Survey Census Data Provider*, and *Reviewer*. In this case, the Summary Files were directly released to NCES for distribution on their web site, not AFF.

2.1.2. DADS Key Functions

DADS provides eight key functions that support DADSO services and the aforementioned systems and actors. These functions, as seen in **Figure 2:** DADS Functional Architecture, frequently rely on the output from another function. Each key function, as described below, provides several capabilities to serve the **End User** or other entity interacting with DADS or to comply with a BOC mandate (i.e., Title 13).

2.1.2.1. Data Processing (ETL)

Data processing, or Extract-Transform-Load (ETL) processing, is performed on all data received by DADS, including metadata. Microdata and aggregated data are also processed with the non-spatial geographic data to associate data to the geography. Processed microdata files (i.e., HDF and SEDF) are provided for tabulation. Processed microdata and aggregated data are loaded into DADS databases for dissemination.

2.1.2.2. Decennial Tabulation

Tabulation is performed on processed microdata files to yield aggregated data (as a dissemination source) or special tabulations. Microdata are aggregated based on predefined calculation rules. The data are then filtered, primarily to comply with Title 13 (Confidentiality) requirements. The resulting data are transformed into data products such as Decennial Census Summary File 1, which require Analyst/Statistician validation.

2.1.2.3. Print Dissemination

DADS provides image-ready PDF files to the product distributor. Print dissemination involved generating predefined reports from the aggregated data then creating Adobe PDF files of the reports.

2.1.2.4. Web Dissemination

DADS provides several functions to support web dissemination. These functions are performed against data products produced from the aggregated data or the processed microdata. Microdata are only accessible via web through ad-hoc processing (see below) or as a sample download file (i.e., public use microdata sample).

DADS presents aggregated data in tabular form or within a map. The tabular presentations are also available for download. To assist with the presentation, DADS provides a geographic selection function to allow users to select their geography or geographies of interest. A search capability is also provided to assist users with obtaining data. Supporting information such as help and technical documentation assists users as well.

2.1.2.5. Ad-hoc Web Processing

Ad-hoc web processing is provided against microdata. Ad-hoc web processing against microdata is protected with confidentiality filters based on rules for electronic disclosure limitation developed by the Census Bureau. The tables that are presented are in a format specified by the user. At the user's request, the ad-hoc aggregation may also be presented as a chart.

2.1.2.6. System Monitoring

System monitoring is performed at three different levels: application usage; data usage; and system performance. Application usage is logged to show users activity within the website including the number of visitors and which pages are most accessed. Data usage is logged to indicate which data products and geographies are accessed by the end users. System performance, such as throughput and availability, is logged for the purposes of verifying system operations.

2.1.2.7. Feedback Collection

DADS collects user feedback to troubleshoot dissemination issues and as a vehicle to send user requests for assistance to customer service. Once the feedback is collected, it is routed to customer service for their action.

2.1.2.8. System Hosting

DADS provides hosting services for prototypes or fully developed systems. The systems (and their data) are deployed in the DADS production environment, using DADS hardware and network components.

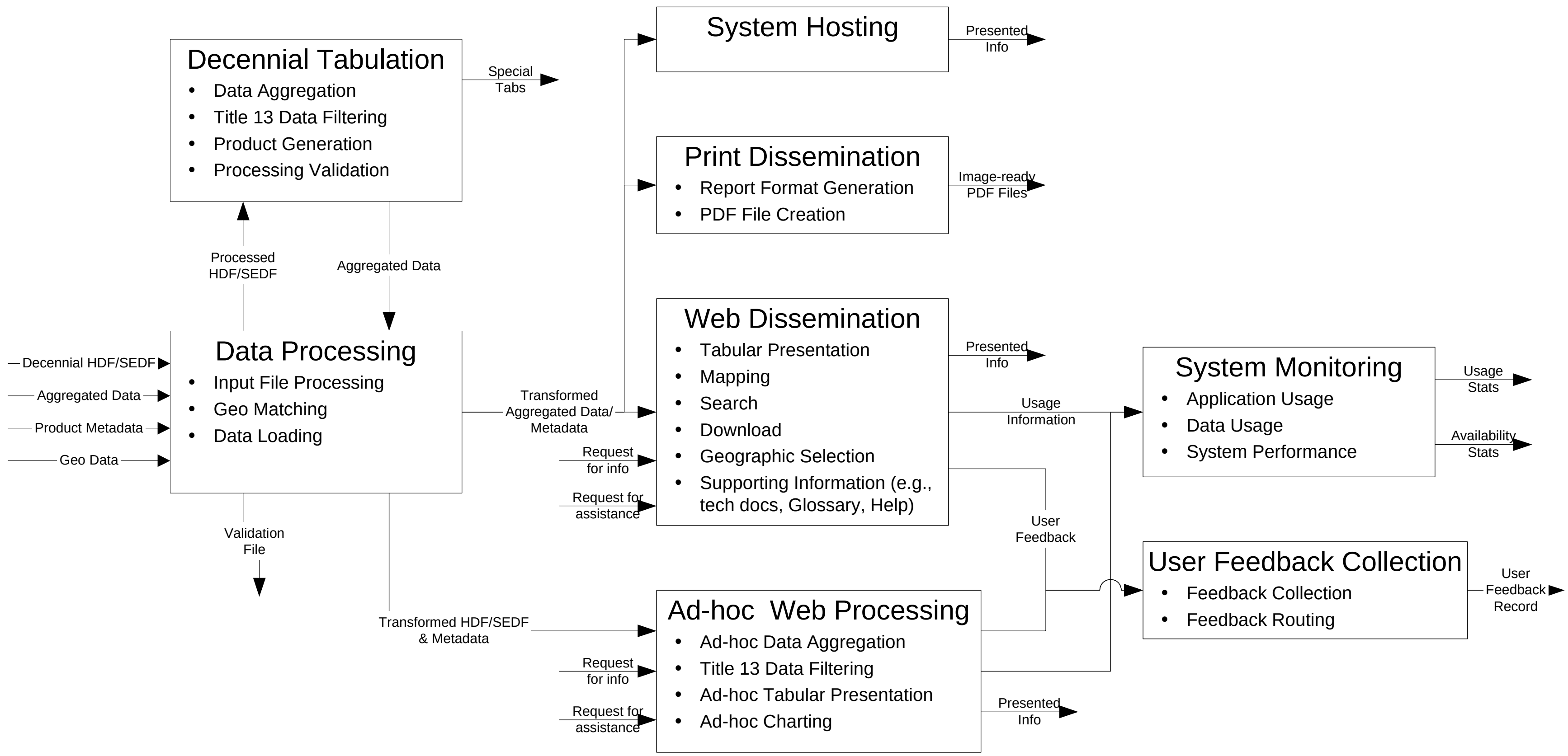

Figure 2: DADS Functional Architecture

2.2. DADS Key Systems

DADS has three key systems that implement its functional capabilities. The functions supported by these systems frequently overlap, and in most cases, were developed separately because of different schedule and business concerns. For instance, data processing is implemented in all DADS systems. For a list of the functions implemented by each system, see **Table 2:** Functional Capabilities Implemented in DADS Key Systems.

2.2.1. American FactFinder

American FactFinder (AFF) is the main vehicle of data dissemination for DADS. The system is a web-based application that is available to the public. Pre-defined data products are presented in tabular or map form.

AFF provides web-based and print dissemination (through the PHC Report System). Since it is publicly facing, usage and performance of the system is closely monitored. Feedback from the system is also closely monitored for issues and routed to the BOC customer service organization, MSO.

2.2.2. Advanced Query

Advanced Query (AQ) is the application that processes user requests for ad-hoc tabulations of the Decennial data. It is accessible on the Web to authorized users only. A BOC Intranet version allows access to the Advanced Query system with the confidentiality filter disabled.

Since AQ provides ad-hoc tabulations, the business rules for aggregating and filtering the microdata are carefully defined by the Census Bureau to ensure the confidentiality of survey respondents. As with AFF, AQ provides functions such as geographic selection and search to aid end users.

2.2.3. Data Products Production

Data Products Production (DPP) is the large-scale tabulation system for the Decennial Census. Its output serves as a source of aggregated data for AFF. It also produces special tabulations.

Functional Capability	AFF	AQ	DPP
Input File Processing	X	X	X
Geo Matching	X	X	X
Data Loading	X	X	X
Data Aggregation			X
Ad-hoc Data Aggregation		X	X
Title 13 Data Filtering		X	X
Product Generation			X
Processing Validation			X
Tabular Presentation	X		
Ad-hoc Tabular Presentation		X	
Ad-hoc Charting		X	
Mapping	X		
Search	X	X	
Download	X	X	
Geographic Selection	X	X	
Supporting Information	X		
Report Format Generation	X		
PDF File Creation	X		
Feedback Collection	X	X	
Feedback Routing	X		
Application Usage	X	X	
Data Usage	X	X	
System Performance	X	X	X

Table 2: Functional Capabilities Implemented in DADS Key Systems

As shown in **Figure 3:** DADS Key Systems, these three systems are relatively independent. Some functional capabilities that overlap between systems are not integrated. For instance, each system has distinct methods, tools, and environments for Data Processing, a key function for on-going data deployments. Other functional capabilities, such as Search, are incorporated within the system's application.

The architecture for each system is also distinct.

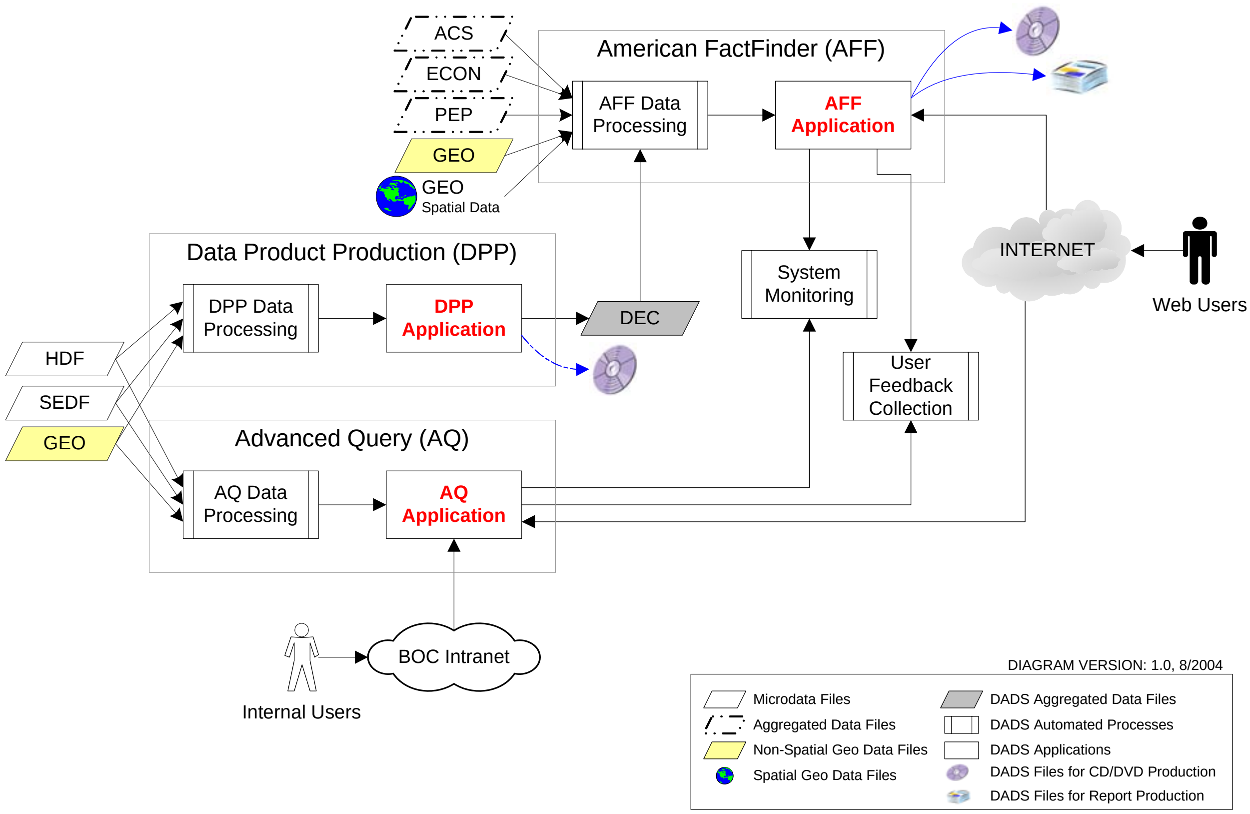

Figure 3: DADS Key Systems

In the following sections, the logical and physical architecture of each system will be described separately. This will show how specific capabilities are implemented within each system given the system's primary purpose, end-users, and related BOC mandates.

3. AMERICAN FACTFINDER (AFF) ARCHITECTURE

The American FactFinder (AFF) architecture is described in this section using Logical and Physical views. The Logical view includes:

- Design Influences
- Application Architecture Overview

The Physical view includes:

- Design Influences
- Physical Overview
- Software Deployment
- Operational Scenarios
- Hardware Architecture
- Network Architecture

3.1. AFF Logical Architecture

American FactFinder (AFF) is a web-based Internet data dissemination system, available at http://factfinder.census.gov, for four major censuses and surveys:

- Decennial Census (1990 and 2000)
- American Community Survey (1996 to present)
- Economic Census and Surveys (1997 and 2002)
- Population Estimates Program (most recent yearly data)

AFF disseminates summarized Census data products in tabular format predefined for each data product. Selected data elements can also be displayed in thematic maps with predefined themes. In addition, geospatial feature and boundary layers, based on shapefiles from Geography Division, are available in American FactFinder. The geospatial layers can appear in dynamically generated thematic maps and reference maps. American FactFinder provides a set of download options to support users who need to extract data for off-line analysis with tools like Excel or SAS. AFF also creates value-added derived products called Fact Sheets that display commonly requested data values into a single presentation.

3.1.1. Design Influences

The vision for American FactFinder from the beginning was always a general-purpose Internet-based data dissemination system. The first Census programs disseminated by American FactFinder were the Decennial 2000 Census and the 1997 Economic Census. Over time, the number of Census programs and datasets has continued to grow steadily.

Based on the enormous size and geographic coverage of the Decennial data products, and the plan to provide on-line mapping, and support flexible ad-hoc user data queries, it was impractical to create a traditional static HTML site with pre-generated pages to meet these needs. Each user could potentially request a unique combination of geographies, tables, or maps. AFF was therefore designed as a **dynamic** application, with each web page created on-demand, based on the user's specific needs. AFF was also designed as a **database** application, so every data element would be retrieved from a relational database.

As part of AFF's initial architectural design in 1998 and 1999, Java and J2EE were chosen as the best available technology for building **dynamic** and **database-driven** web applications on the UNIX platform.

J2EE also provides infrastructure for building highly **scalable** and **fault-tolerant** Internet sites that can support thousand of users.

Another design influence was the expected unprecedented volume of users attracted to on-line Census data. Before AFF, Census data was distributed primarily in printed form to Federal Depository libraries, on CD-ROMs, and on magnetic tapes. The AFF application was designed for:

- Scalability – so additional capacity could be added, as needed, and so the site could handle an expected load of 3600 concurrent users
- Availability – so the site was available 24 hours a day (with provisions for scheduled downtime and maintenance)
- Redundancy – to reduce the number of "single points of failure" in the system; so a single system failure does not impact the site's availability
- Failover – part of redundancy; the infrastructure should detect the failure of key systems, and failover to redundant systems until the failed systems are repaired and brought back online.
- Extensibility – the application and data architecture should accommodate new Census programs and datasets with minimal effort:
    - The application architecture should be metadata-driven, so new datasets can be added to the application with minimal programming changes; the user interface navigation logic and presentation rules should be encoded in metadata, and dynamically discovered by the application.
    - The data architecture should model Census entities like geography, datasets, and tables in an extensible and forward-looking manner, so that future programs and presentations formats can build on the data architecture

Another key design influence that drove the J2EE choice was the requirement for a highly stateful application. The Census Bureau wanted the AFF user experience to mimic a desktop application that "remembers" user preferences and selections as they navigate the application.

The IBM WebSphere Application Server platform and IBM Edge Server were determined to meet all these requirements.

In summary, AFF is based on a number of best-practice architectural principles for scalable Internet applications:

- Distribute user requests to a cluster of fault-tolerant servers using load balancing,
- Use a J2EE application server (WebSphere Application Server) to provide the infrastructure and programming model for building an object-oriented, component-based web application,
- Base the AFF application architecture on the standard model-view-controller (MVC) design pattern, a proven design approach for building web applications,
- Build AFF as a metadata-driven system so the business rules are in the database, not the application, and have the application dynamically discover the business rules.

3.1.2. AFF Logical Application Architecture

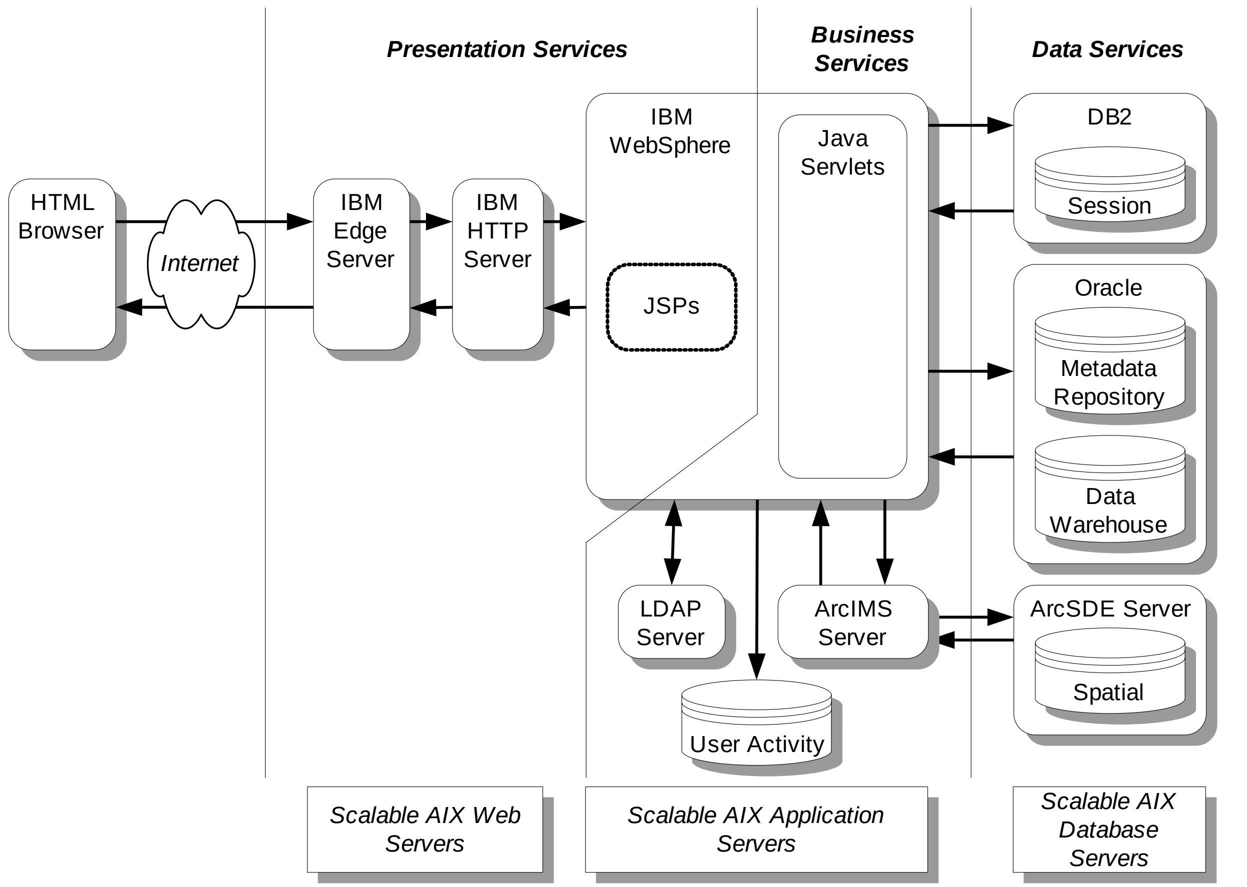

Figure 4: AFF Logical Architecture Overview

Users create and execute queries interactively within an HTML Browser via a series of requests to AFF. These requests are received by the IBM Edge Server and then redirected to an IBM HTTP Server that forwards the request to IBM WebSphere for processing. Custom Java Servlets are executed by IBM WebSphere to process the request and build the response using Java Server Pages (JSPs) technology. The Java Servlets process the user request, and dynamically query the AFF Metadata Repository to build queries against the tabulated data stored in the AFF Data Warehouse.

To improve performance and reduce network traffic, the Java Servlets store the user's Session context in a DB2 database. For user requests involving spatial data, such as reference maps or thematic maps, the Java Servlets send spatial requests to the ESRI ArcIMS Server for processing; the ArcIMS Server uses Spatial data stored in a database controlled by the ArcSDE Server.

For the internal review system used by BOC Subject Matter Experts (SMEs), access is controlled on a Java Servlet by Java Servlet basis using a set of users and groups defined in an LDAP server.

3.1.2.1. IBM WebSphere Edge Server

The **IBM WebSphere Edge Server** includes a component known as **Network Dispatcher** (**ND**). ND helps reduce Web server congestion, increase content availability, and provide improved Web server performance through **load balancing**. ND provides automated routing of web requests to available web servers for balancing or failure recovery.

Network Dispatcher is configured to return a user back to an HTTP/WebSphere Server node based on their IP address. The benefits of returning a user to a node include the reuse of application cache, user session and database resources already in memory to support that user.

IP address-based routing, or "sticky bit" usage, can be problematic because some Internet Service Providers, like America Online, have many users share a fixed pool of IP addresses. Although this may have a small *performance* impact on AFF (because the application caches are not used to their full extent), it has no *functional* effect on AFF, because AFF tracks users by a unique session identifier, not their IP address.

3.1.2.2. IBM HTTP Server

The **IBM HTTP Server** serves static HTML pages and images, and forwards dynamic page requests, via a **HTTP plug-in**, to IBM WebSphere Application Server. The IBM HTTP Server also serves map images generated on-demand by AFF's mapping system.

All HTTP page requests, static or dynamic, are logged in the IBM HTTP Server web server logs.

IBM's HTTP Server is based on the open source Apache web server.

3.1.2.3. HTTP Plug-In

The **HTTP Plug-In** routes requests for dynamic pages from the HTTP Server to WebSphere Application Server. The HTTP Plug-In also performs **workload management** and uses a weighted round robin technique to distribute application server requests (that do not belong to an existing session) across web containers running within an application server node.

After a session is created, all requests associated with the session are directed to the same application server that created the session. For American FactFinder, this is accomplished via a **session identifier** embedded in a HTTP request header. If the application server associated with a session is not available, the HTTP Plug-In will direct requests to an available application server within the cluster. Any session information for the user is recovered from the **persistent session database**, which allows application servers within the same cluster to share session information.

3.1.2.4. WebSphere Application Server

The dynamic component of the American FactFinder application is a J2EE application built using **IBM WebSphere Application Server** (**WAS**). WAS version 5.1 is an application server based on **J2EE** version 1.3.

American FactFinder is a custom Java application that uses the J2EE infrastructure components **Java Servlets**, **JavaServer Pages**, **JDBC**, and **XML** to build and deploy a robust and scalable enterprise web application.

3.1.2.5. Application Server Clones

The production American FactFinder system is configured as a multi-node WebSphere **cluster**. A cluster is a collection of identically configured nodes running the AFF application. A **node** is a physical machine. Running multiple nodes provides AFF with **horizontal scalability** – performance can be enhanced, if necessary, by adding more nodes to the cluster.

A given node can run multiple application servers. Each application server is called a **clone**. A **clone** is a stand-alone Java Virtual Machine (JVM) that runs a complete, standalone instance of the AFF application. Clones provide AFF with **vertical scalability** – the ability to fully utilize the available memory and processing power on a single node.

All clones in the WebSphere cluster share a common session database, which provides session failover in the event a clone dies between user requests.

3.1.2.6. Session

The HTTP protocol between a web browser and web server is stateless, which means each connection is made without any reference to previous connections. Session is a technique that overcomes the stateless nature of the HTTP protocol, and allows a user to engage in a “stateful” conversation with an application.

American FactFinder uses the HTTP **Cookie** method to transfer the **session identifier** on each HTTP request from the browser to the server. Technically, AFF uses only **transient cookies**, which are valid for the duration of the user’s browser session; they are not saved to the user’s local browser Cookies folder. The session identifier is used to accumulate state information on the server in a **session object** without transferring that information with every page request.

The **session object** is a Java object provided as part of the J2EE infrastructure. The session object is stored in a persistent **session database** shared by all the application servers in the WebSphere cluster. At the end of each servlet call, if the session object has changed, it is converted to a string (in a Java process called **serialization**) and written as a record in a **persistent session database**, indexed by the session identifier.

For AFF, the **session object** tracks state information for each **context** in the American FactFinder application (Quick Tables, Geographic Comparison Tables, Detailed Tables, Thematic Maps, Reference Maps, Custom Tables, etc.)

For each context, the session object stores the user’s selected dataset, tables, themes, geographies, geographic components, characteristic iterations, etc. When a user navigates to a context previously visited in the same session, AFF attempts to restore the user’s previous settings if possible.

3.1.2.7. Session Database

To provide application failover for stateful applications in a clustered environment, IBM WebSphere Application Server supports a **persistent session database** as part of the core J2EE WebSphere infrastructure.

For every user request, the application server clone extracts the session identifier from the HTTP request header, and either creates a new session (if there’s no cookie), or retrieves the session object from in-memory cache, or the session database. Any changes to the session object are written back to the session database

The session identifier encodes the WebSphere node and clone name that served the page. IBM Edge Server and the HTTP Plug-In work together to make a best effort to return a given user to the same server (using Edge Server’s IP address **server affinity**) and same clone (using the HTTP Plug-In’s workload manager that uses the session identifier to achieve **clone affinity**) for each page request.

When a user exits their browser, the browser cookie is discarded by their browser, and the corresponding WebSphere session object is orphaned. A separate clean-up process removes discarded session objects from WebSphere memory and the DB2 session table after a certain period of inactivity (controlled by the WebSphere administrator).

3.1.2.8. IBM LDAP Server

On the AFF Internal Review environment, AFF is password protected and available only to authorized users. The extra security is required since the AFF Internal Review environment may contain data under review by the data provider and not publicly released. Security is implemented with standard WebSphere LDAP security and uses the IBM Directory Server to manage users and groups. The WebSphere Administration Client is used to assign user and group privileges to specific servlets in the AFF application.

WebSphere security is not used on the AFF production system.

3.1.2.9. ArcIMS

American FactFinder's mapping subsystem is implemented with ESRI's ArcIMS and ArcSDE COTS. **ArcIMS** is an Internet map-publishing engine that provides on-demand thematic and reference map services, and geocoding services to AFF.

Geocoding converts an address into an (x,y) coordinate or a Census Block ID. Geocoding supports the "Search by Address" functionality in American FactFinder.

WebSphere and ArcIMS interact via the **ArcXML** language. WebSphere mapping requests, packaged as ArcXML messages, are sent to ArcIMS using the **ArcIMS connector**. The **ArcIMS connector** is Java library that allows WebSphere and ArcIMS to exchange ArcXML messages. A single map request may require several ArcXML request/response interactions between WebSphere and ArcIMS.

ArcIMS manages a set of **Spatial Servers** that provide query, geocoding, and image services. The query and image services use ArcSDE to resolve geospatial queries or retrieve features and boundaries. The geocoder uses a database file from Sagent. Sagent is a third-party company whose geocoding technology was integrated into ArcIMS as part of the custom work done on the AFF mapping system.

A subject matter expert creates ArcXML map configuration files, called **axl files**, with an XML Editor like XML Spy or ArcIMS Author, which define a default set of instructions for map properties and rendering. At startup, ArcIMS loads the **axl files** to create image services in the spatial servers.

The image server builds maps as GIF files and stores them on a file system available to the IBM HTTP Server. ArcIMS returns to WebSphere not the image, but rather a pointer to the image on disk. In turn, AFF returns to the user an HTML page with a link to the map image.

3.1.2.10. ArcSDE

ArcSDE is an ESRI application that provides database-independent geospatial functions on top of a relational database called a **geodatabase**. AFF uses Oracle to store the ESRI geodatabase. All interactions with the geodatabase are done via ArcSDE, so the geodatabase is considered a proprietary vendor database.

3.1.2.11. AFF Application Architecture

AFF uses a "**Model 2**" application architecture, the most widely used architecture for web-based applications. Model 2 is also called the **model-view-controller** (**MVC**) architecture. MVC is a layered architecture that cleanly separates areas of concern in the application:

- The **model** contains the **business logic**. The AFF model is implemented as several layers – the **service**, **factory**, and **builder** layers work together to create AFF **business objects**.
- The **view** contains the **presentation logic** or **user interface** and is responsible for rendering model objects into appropriate HTML and JavaScript. The AFF **view** is implemented using JavaServer Pages.
- The **controller** contains the application navigation logic and coordinates between the **model** and **view**. The AFF **controller** is implemented using servlets.

By decoupling the business and presentation logic into separate layers, namely the **model** and **view**, an application is easier to maintain and enhance.

3.1.2.11.1 Java Server Pages

Java Server Pages are part of the J2EE technology stack, and allow programmers to create HTML-like pages that contain Java code fragments. The JSP pages are compiled by the application server and operationally behave just like a J2EE servlet. AFF uses Java Server Pages exclusively for the final presentation of a dynamic page to the user.

3.1.2.11.2 Design Patterns

Design patterns are proven reusable solutions to common problems. They are well documented in the programming community and provide a common language for designers and architects to discuss recurring problems and proven solutions.

The model-view-controller architecture is one of the best-known **design patterns.** AFF uses a number of design patterns in its design, including thread and object pools, singletons, producer-consumer threads, and object factories.

3.1.2.11.3 MVC Controller

In AFF, the MVC **controllers** are implemented as servlets and represent the public interface to AFF. The controller coordinates the flow of information between the **model** and **view**. The controller has the following responsibilities:

- Validate the user's HTTP request and extract the user's query parameters from the HTTP message
- Update or create the session object with the user's request
- Parse the user's request, and make one or more calls to the **service** layer to assemble the **business objects** to respond to the user's request
- Bundle the **business objects** returned from the service layer into a **view adapter** object, and forward the view adapter to the **view** for presentation.
- Log the details of the user's request into a XML file for later loading in the **User Activity database**
- If necessary, handle any unexpected errors conditions and display the AFF error page

3.1.2.11.4 MVC View

In AFF, the MVC **views** are implemented as JavaServer Pages to render the HTML screen to the user. Every JSP is provided with a **view adapter** – a Java object created by the controller that passes the business objects created by the service layer. The JSP is responsible for rendering the business objects in the view adapter on the page in a manner consistent with the AFF look and feel.

AFF JSPs may be built for smaller JSP fragments like the header, footer, and left navigation. The JSP fragments are common UI elements that appear every on every page.

3.1.2.11.5 MVC Model

In AFF, the MVC **model** is implemented as a three separate layers – the **service**, **factory**, and **builder**. The layers work together to create business objects for the controller.

3.1.2.11.6 Service

Based on the request from the **controller** layer, the **service** layer creates AFF **keys** and invokes AFF object **factories** to build **business objects**. The AFF key acts as the "primary key" for the business object. Every business object is uniquely identified an AFF **key**. The business objects are returned to the controller.

3.1.2.11.7 Factory

AFF uses the **factory** design pattern to create AFF business objects. The **factory** design pattern allows users to create business objects without understanding the details of their construction from Oracle. The AFF key contains the information the factory needs to create the object.

To create a business object, the factory does the following:

- Validate the AFF key object and make sure it contains the necessary information

- If the factory is configured as **caching**, it checks for the requested object in its object cache. Objects are indexed by key.
    - If the object exists in cache, it's returned to the caller
- If the object doesn't exist in the cache, or if the factory is configured as non-caching, a request to build the object is delegated to an AFF **builder**. Before returning the business object to the caller, a caching factory stores the object in cache, indexed by key.

3.1.2.11.8 Factory Caching

AFF contains a custom-built **object-caching** layer. Factories can be configured on a case-by-case basis to **cache** the objects they create. Caching makes sense for business objects that are expensive to build in terms of database resources. The caching subsystem is highly configurable and allows the WebSphere administrator to control the object aging policy and maximum object cache size on a per-factory basis. In low-memory situations, cached objects can be deleted to free-up memory.

Each AFF object cache is defined as "serializable." A serializable object is one that can be written to and read from disk. At startup, AFF looks for serialized cache files on disk to quickly re-instantiate the expensive in-memory object caches.

The WebSphere administrator manages the creation and loading of the AFF object serialization files. Serialization files can become stale if the business objects they represent are updated in the underlying Oracle database. Therefore the serialization files are routinely deleted once a day, or immediately after any database changes.

3.1.2.11.9 Builder

The AFF **factory** delegates the work of creating objects to the AFF **builder**. Using the AFF key, the **builder** creates **business objects** by issuing SQL queries against the AFF data warehouse or metadata repository. Generally, the SQL needed to populate a business object is embedded directly in the builder java code, as opposed to a stored procedure. After the SQL query is executed, the builder instantiates the requested AFF object(s) and returns them to the factory.

Builders are the only AFF components with direct knowledge of SQL. Builders use WebSphere connection pools to acquire and release database connections. WebSphere's database connection pooling is part of the J2EE infrastructure and form AFF's **data access** layer.

3.1.2.11.10 Business Objects

Business objects (also called **domain objects**) represent the business entities within AFF. A business object may represent a single row, or many rows, in an Oracle table, or may represent a complex join of multiple tables in Oracle.

There are approximately 100 business objects in AFF (the number grows as new presentation formats and functions are added). Some example business objects in AFF, each implemented as a Java class:

- **Dataset** – represents a Census dataset,
- **DataItem** – represents a data item in a demographic table or an Economic dataset table,
- **GCTRowStub** – represents the "row stub" portion of geographic comparison table,
- **GeographyBucket** – represents the geography for a specific year and program,
- **MatrixTable** – represents the most detailed tabular data from a data provider
- **JamValue –** represents a "jam value" or substitution value that replace one value in a table cell with another value

In summary, a business object usually represents an object (like a Matrix Table) or part of an object (like a Jam Value) visible to the end user. A **business object** is created by a **builder**, based on a **key**, and managed by a **factory**. Business objects themselves know nothing about Oracle or relational databases. The **service**, **factory** and **builder** layers define AFF's MVC **model** layer.

3.1.2.11.11 Data Access

AFF uses WebSphere connection pooling to manage its database connections with Oracle and DB2. At startup, WebSphere allocates a fixed number of connections to each database. The WebSphere administrator controls the size of each pool. The connection pool size is a key tuning parameter for the AFF system.

Generally, *creating* and *deleting* a database connection is much more expensive operation than actually *using* the connection. Pooling database connections eliminates the need to constantly create and delete connections by creating a finite pool of reusable connections that are created once and never deleted.

The AFF builder *acquires* a connection from the pool, uses it to issue a SQL query to Oracle, and then *releases* it to the pool. If all the connections in the pool are in use, new requests block waiting for an available connection. Under heavy load, if all the connections are in use, users will experience slowness in AFF.

It's critical for the performance of interactive web-based database applications like AFF that database queries be fast – long running SQL queries quickly exhaust the pool and degrade system performance.

3.1.2.12. Custom Logging

AFF logs a detailed XML formatted record to a file for every **result page** it serves (a **result page** is defined as a page that serves a table or map to the user). To save disk space, AFF does not log user **navigation pages** like "select geo" or "select table". The AFF XML logs provide more detail about the user's request than can be captured in the HTTP web logs. Nightly, the XML logging records are loaded into a **User Activity database**. Reports generated on the User Activity database provide detailed insight into how users access American FactFinder.

3.1.3. AFF Logical Data Structure

There are a total of eight logical data stores associated with the AFF production system. Note that some data stores may live in the same physical database, but under different schemas for logical separation. Specifically, the User Activity database resides in the Metadata Repository database under a separate schema.

The **Metadata** databases contain data that support the AFF user interface and metadata about tabular and geospatial data:

1. English Metadata Repository (EMDR)
2. Spanish Metadata Repository (SMDR)

The **Data** databases contain tabulated demographic and economic data, and geospatial data from the Census TIGER system:

3. Data Warehouse (DW) for tabular data
4. Spatial Data Warehouse (SDW) for geospatial data

The **WebSphere Administrative and Infrastructure Support** databases support the WebSphere runtime environment:

5. WebSphere Administration Database (ADMIN)
6. Session Database (SESS)

The **Feedback** database is a write-enabled database that captures user online feedback messages:

7. Feedback Database (FB)

The **User Activity** database is a write-enabled database used for offline batch loading of the user activity records logged daily by AFF:

8. User Activity Database (UA)

All of the databases are Oracle, with the exception of the **WebSphere Administrative and Infrastructure Support** systems, which run DB2. The DB2 database is provided as part of the WebSphere Application Server and is the recommend database for these administrative and support tasks.

3.1.3.1. Metadata Repository (MDR)

AFF uses the metadata stored in the AFF Metadata Repository (MDR) to control user navigation and for data presentation. The MDR is an **Oracle** database. There are two MDRs, one for English-language content and another for Spanish-language (SMDR) content.

The metadata is also a "roadmap" for finding tabular data within the AFF Data Warehouse (DW).

The data stored in the Metadata Warehouse is organized around several major functional groups:

- **Meta Central** – Common metadata for all subject areas and configuration & control metadata
- **Data Sets** – Metadata for the products and navigation paths available for a data set
- **Map View Group Management** – Metadata to support the user navigation of maps, and to present meaningful combinations to the user
- **Matrix Tables** – Metadata for the detail tables in the Data Warehouse
- **Virtual Tables** – Metadata for tabular derived products within the Data Warehouse
- **Non Spatial Geo** – Metadata for geographic navigation and selection
- **Survey Products and Questionnaires** – Tier 1 products including PDF files and survey information
- **Thematic Maps** – Metadata for thematic maps
- **Deliveries and Loads** – Metadata associated with the processing of data and metadata
- **Econ** – Metadata to support user navigation of the detailed datasets for the 1997 and 2002 Economic Census and surveys.

3.1.3.2. Spanish Metadata Repository (SMDR)

Structurally, the Spanish Metadata Repository (SMDR) is identical to its English counterpart. The main difference is all the metadata visible in the AFF user interface have been translated into Spanish. The SMDR supports the Spanish component of the AFF web application, which presents the user interface in Spanish, and highlights only the Decennial 2000 Census data for Puerto Rico. Only metadata for Puerto Rico is loaded in the SMDR.

There is no separate Spanish Data Warehouse. The data for Puerto Rico was tabulated as part of the official Decennial 2000 Census data products, and resides in the AFF Data Warehouse. The Puerto Rico data is available in both the English and Spanish version of AFF.

3.1.3.3. Data Warehouse (DW) for tabular data

The AFF Data Warehouse (DW) is an **Oracle** database used to store large quantities of tabulated demographic and economic Census data. Unlike a star schema, there are no dimension or fact tables in the DW. Each tabular table resides in a separate Oracle table. The presentation instructions for each table are contained in the AFF Metadata Repository. This decoupling of data and presentation into separate databases is part of AFF's overall metadata-driven approach.

Each DW table represents a tabulated table for a BOC program, survey, year and product combination. The AFF Data Warehouse (DW) contains the following objects:

- **Matrix tables** for Demographic surveys (several thousand tables). Matrix tables are indexed by GEO_ID (or GEO_ID and characteristic iteration for iterated products). The program, survey, survey year, dataset, and matrix name are encoded in the table name (e.g., DEC_2000_SF1_U_H003). The matrix tables are the source for AFF's demographic detailed tables, quick tables, and geographic comparison tables.

- **Dataset tables** for Economic surveys (several hundred tables). The structure of the Economic datasets is highly variable, depending on the information collected for that industry. The structure of each dataset table is defined in the AFF Metadata Repository.
- **SQL database views** for Demographic and Economic derived products. The demographic derived product views are created by the AFF **Spec Processor** tool, and are created as part of the overall process of creating derived products in American FactFinder. The Economic derived products (Geographic Quick Reports, Industry Quick Reports, and Product Quick Reports) are created with a separate process by the DADS staff, and not with the Spec Processor,
- **Copies** of six **MDR tables** are in the DW to improve the performance of the AFF application. Local copies avoid costly joins between the DW and MDR databases. During the final rendering of a table, the AFF application has to join data from the DW with the MDR; these local copies significantly improve the performance of this task.
- A small collection of database **stored procedures**, functions, and custom object types to support the Economic Census. Generally in AFF, the SQL code is embedded in the Java application, not the database. These stored procedures are the exception to that rule.

3.1.3.4. Spatial Data Warehouse (SDW) for geospatial data

The Spatial Data Warehouse (SDW) is an **Oracle** database that stores shapefiles from Geography Division (GEO). Spatial files store coordinates that describe the location and shape of geographic features, thereby allowing for their depiction in map form. AFF's spatial database supports the following AFF functionality:

- Map-based selection of one or more geographies as part of a data query,
- The creation of reference maps for identification of survey-specific geographies,
- The creation of thematic maps to aid data visualization.

The spatial file deliveries from GEO are defined by detailed specifications written by DADS. The files are delivered as shapefile (an ESRI file format) and loaded into the spatial data warehouse using ArcSDE. There are several types of features managed by ArcSDE:

- Survey-specific tabulation geography boundaries – specified by DADS spatial file specifications to GEO;
- Yearly detailed feature networks – also specified by DADS spatial file specifications to GEO; features represent common, orienting symbols like roads, railroads, streets, etc. that can appear on maps to orient the user.

3.1.3.5. WebSphere Administration Database (ADMIN)

The WebSphere Administration Database (ADMIN) is a **DB2** database that stores the configuration information for the WebSphere cluster and the application servers it contains. During WebSphere startup, this database is used to instantiate the WebSphere runtime environment and creates the application server clones. All administration on this database is done through the WebSphere Administration Client, a standalone Java application.

3.1.3.6. Session Database (SESS)

The session database (SESS) is a **DB2** database used by the WebSphere J2EE infrastructure to store, retrieve, and delete serialized user session objects. The session objects are stored in a simple DB2 table, with fields that track the session "create" and "last update" date and time. The session object is stored as a BLOB (binary large object) field. The session identifier is the primary key for the session table.

Periodically, each WebSphere clone executes a process to delete abandoned session objects from the database. Users who attempt to access an abandoned session object receive the “Session Timeout” error page.

All of the clones in the WebSphere cluster write to and read from the same session database. A separate database connection pool, configured by the WebSphere administrator, is used to control access to this database.

3.1.3.7. Feedback (FB)

AFF user feedback message are stored in the Feedback (FB) **Oracle** database. The feedback record contains all the fields on the form (feedback, name, email address, phone number), as well as additional fields, not visible to the user, that track the page on which the error occurred, the content of the session object, and any Java error message. This extra information assists the AFF technical team to debug user problem reports.

The Feedback records are periodically transferred to the AFF Internal Review system for further analysis and processing, and also shared with the RightNow system.

3.1.3.8. User Activity (UA)

The User Activity (UA) Oracle database contains detailed information about every result page served by AFF. (A result page is defined as a page that displays a table or map.) These detailed logs capture more information than is possible in the HTTP web server logs. Specifically, the following information is captured from the user’s session, for every result page:

- The session identifier,
- If the request was a download, the download format used (Excel, TXT, etc.),
- The dataset and geobucket,
- The list of geographic summary levels, and a count of how many instances of each summary level,
- The list of tables and characteristics iterations,
- If a ECON request, the Economic dataset name,
- If a map request, the GEO_ID, theme, and service name used to create the map.

3.2. AFF Physical Architecture

AFF's physical architecture must be able to support a large number of concurrent users with minimal downtime and reasonable response times.

3.2.1. Design Influences

As an interactive web application, AFF must make available large volumes of tabulated census data and maps to thousands of concurrent users with quick response times. Some of the significant design influences include:

- AFF must support 3600 concurrent users.
- AFF is a “real-time” read-only system and must have reasonable response times, at most 5 to 7 seconds for most interactions, and up to 15 seconds for maps. In addition, the system is expected to have minimal maintenance downtimes.
- AFF is used by “the general public” without authentication and must scale on-demand to handle a sudden flood of users. This scaling includes processing and network capacity. AFF is expected to shunt users gracefully if the number of concurrent users would significantly degrade AFF system performance.

- AFF must be "self-healing" and continue to operate in the event of a subsystem failure.
- The number of programs, surveys and data sets disseminated by AFF increases over time so the AFF system architecture must support scalability of processing units, storage units and network bandwidth.
- AFF must support separate development, product assurance, internal review and production environments.

3.2.2. AFF Physical Overview

Powerful, multiprocessor AIX systems defined as a SP cluster with multiple nodes are used by the AFF production system to provide the computing power needed to handle a large number of users. The high-performance database system Oracle is used to store and manage the large data sets associated with AFF. The DB2 database system provides infrastructure and administrative support for WebSphere.

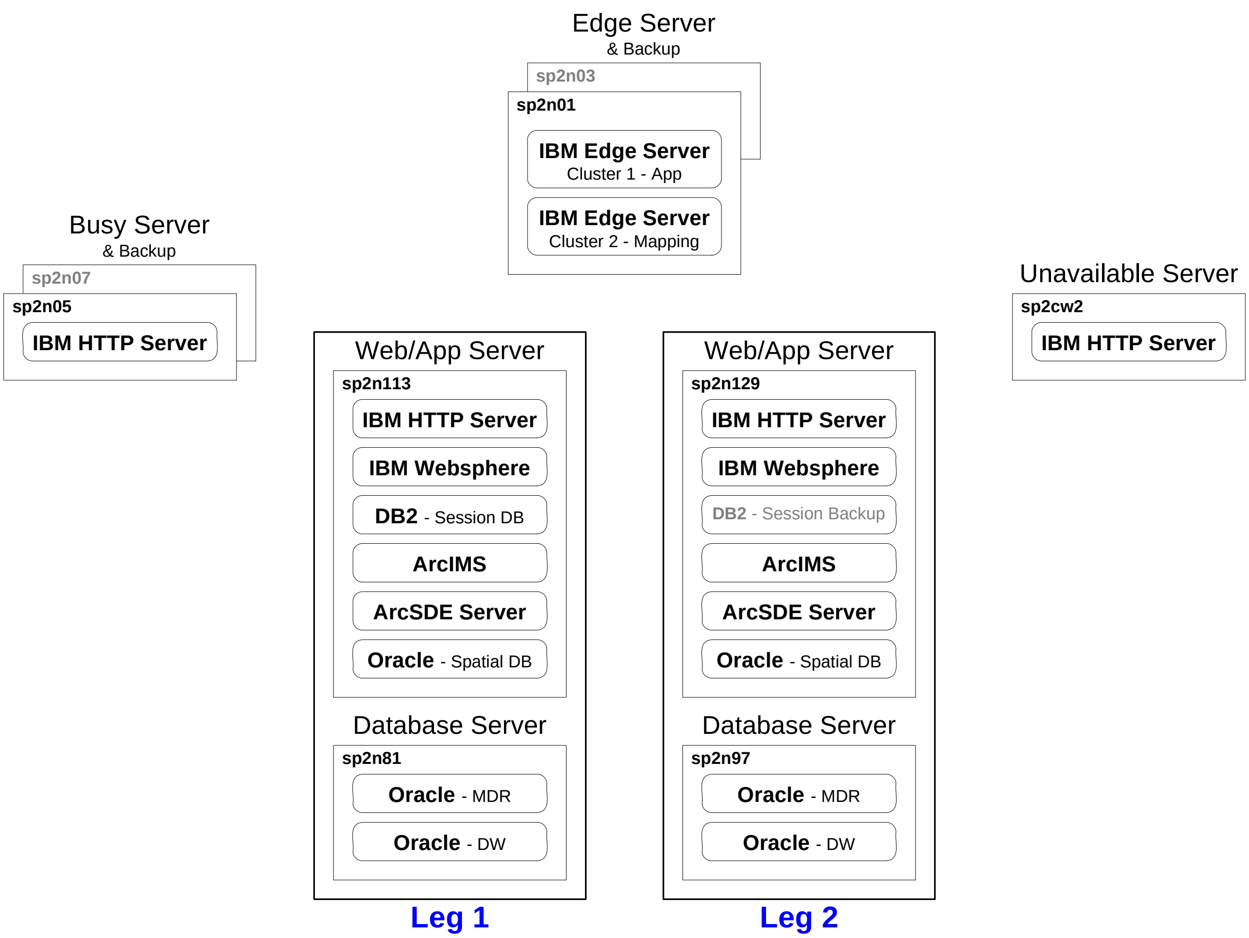

Figure 5: AFF Hardware Overview

The AFF production system is configured into two identical systems, called Leg 1 and Leg 2, to provide redundancy and enhance performance. Each leg is independent of each other, installed on separate SP nodes. Redundancy within each leg is provided by parallel systems or backup systems (hot-spares).

Each leg contains independent HTTP Servers, WebSphere and ArcIMS servers; multiple instances of WebSphere and ArcIMS are used to distribute the load within a leg and across the processors of the SP nodes.

Each leg has its own independent read-only copy of the metadata repository and data warehouse on separate Oracle database servers. Likewise, each leg has a separate copy of the spatial database managed by a separate ArcSDE instance. The session database must be shared between both legs using a single DB2 database server; a backup DB2 server ready to take-over provides redundancy if the primary fails.

During normal operation, the Edge Servers redirect incoming requests among the legs to balance the load on the web and application servers. When a leg is down for planned maintenance, upgrades, or an unscheduled outage, the Edge Servers redirect all incoming requests to the other leg. The Edge Server is configured to be "IP sticky" – it routes a returning IP address to the same WebSphere node. New IP addresses are routed to the least busy application server. This node-affinity leverages the session and object cache in WebSphere. Furthermore, if WebSphere is configured with multiple JVM clones, the WebSphere workload manager routes returning user to the same JVM instance to leverage the session and object cache.

When the incoming requests would overwhelm AFF, the Edge Servers redirect a subset of the incoming requests to the Busy Servers which returns a "System is busy, please try again later" message to users.

When both legs are unavailable, the Edge Servers redirect all incoming requests to the Unavailable Servers which returns a "System is unavailable, please try again later" message to users.

The Edge Servers are also used to distribute and balance map requests among the mapping subsystems on both legs.

AFF hardware is utilized to the fullest extent possible. The powerful SP systems perform multiple tasks on the same hardware; for example, web servers and application servers execute on the same nodes.

3.2.3. AFF Software Deployment

AFF software is divided into three categories[1]:

- **Custom** – Applications and databases written to support a system function or enhancement
- **COTS** – Vendor-produced software, also known as Commercial Off-The-Shelf (COTS) software
- **Server** – Applications supporting the operation and administration of the servers and network

The following table is a high-level matrix of the AFF software and databases grouped by the above categories:

	Software	Description
Custom	AFF App (aff_app_1.war)	AFF application written in Java and delivered as a **jar** file. Additional jar files with 3rd party tools used by AFF are also installed as part of the AFF App. The jar files contain all static content (HTML/CSS/etc) as well as the objects resulting from compiling the Java code. Java code includes both Java classes and servlets, as well as JSPs. The **web.xml** file is used to map named servlets passed in URLs to a specific Java servlet.
	English MDR	English language metadata repository database. This normalized database contains all language-specific navigation information.
	Spanish MDR	Spanish language metadata repository database. This normalized database is a subset of the English MDR translated into Spanish; the Spanish MDR only covers Puerto Rico.
	Data Warehouse DB	Statistical tables database. This denormalized database contains only the tabular data values. The MDR is used to determine which rows and columns are used to build result products.
	Spatial DB	Spatial (geographic) data and mapping support database. This database includes spatial data for tabulation geographies and orienting features (such as roads and railroads).

[1] Note this list does not include the programmer development environment. That software list is covered separately in the AFF Detailed Design documentation. That document also discusses AFF's supported target web browsers, and the emphasis on support for web standards to reduce AFF's dependency on specific browser platforms and versions.

	Software	Description
	User Statistics DB	Results page and user request statistics database. AFF logs records describing the query for every result web page; this data is processed nightly into a normalized database that is used to build user activity reports.
	Session DB	User session persistence database. This database, common to both legs, allows users to maintain their session even if the leg handling their requests becomes unavailable. This database is also used by AFF to store session parameters associated with each major navigation path to avoid having to pass the entire transaction state with each request/response message.
COTS	IBM Edgeserver v5.0.2.13	Load balancing server that distributes requests between the legs. The edge server is normally run in "sticky-bit" mode so that all requests from the same IP address are handled, if possible, by the same leg. This is done for performance reasons.
	IBM HTTP Server v1.3	Service HTTP requests, and forward dynamic page requests to WebSphere. The IBM HTTP Server is an enhanced version of Apache, the most successful and popular web server.
	IBM WebSphere v5.1.2	Family of Java development and web application server product. WebSphere provides the Java Virtual Machine (JVM) within which AFF App executes. Multiple instances of WebSphere are run on each leg for maximum performance.
	ESRI SDE v9.1	Spatial database manager designed to work with ArcIMS.
	ESRI ArcIMS v9.1	Map creation engine and search-by-address engine. ArcIMS creates the maps based upon XML requests from AFF App using a plug-in API provided by ESRI. Multiple instances of ArcIMS are run on each leg for maximum performance and to prevent the mapping system from being a bottleneck. The edge server is used to distribute requests from AFF App/WebSphere to the various ArcIMS instances. Whenever possible, map requests are handled with the same leg for performance reasons. Maps created are written to disk as static files to be later retrieved by the user's browser. The file systems where maps are written is mounted on both legs. AFF caches the most frequently used maps but always creates a map if a cached copy cannot be found.
	Sagent Address File	Address data file and supporting libraries used by the Search By Address function. This file is accessed by a plug-in into ArcIMS. The data file is delivered on a periodic basis by Sagent (a third party geocode vendor), although not all updates are released on AFF.
	IBM DB2 v8.1	Database /management system to support the Session DB. Because the session information must be available to both legs, the same DB2 instance is used by both legs. A backup DB2 database is available on a hot-spare in case the primary DB2 database becomes unavailable.
	Oracle RAC v9.2	Database system – supports multi-node parallel storage clusters. There are two independent instances, one for each leg. The data is stored on the ESS system to ensure data availability.
	LDAP Server v5.1	Server used to authenticate access to Internal Review (IR) environment. IR is used by subject matter experts to review data dissemination data sets before general release; IR therefore is password protected to avoid unauthorized users from gaining access to pre-release data.
System	AIX v5.1	Server operating system. AIX is a high-performance operating.
	PSSP v3.5	Parallel System Support Program for AIX OS.
	TSM Server v5.1	Tape back-up system server used to backup all systems nightly to a tape library.
	TSM Client v5.1	Tape back-up system client program that runs on each system and used by the TSM Server to access the files on the system.
	Concord eHealth v5.6	Web-based performance monitoring tool that reports the activity associated with system resources and activity including CPU, Disk, Memory, Processes, and Network. This monitoring sends the data to a remote system for processing; as a result the load on the monitored system is minimal.

Table 3: AFF Production Software

The following figure shows the distribution of the above software on the AFF production hardware in order to meet the requirements for availability, self-healing and on-demand.

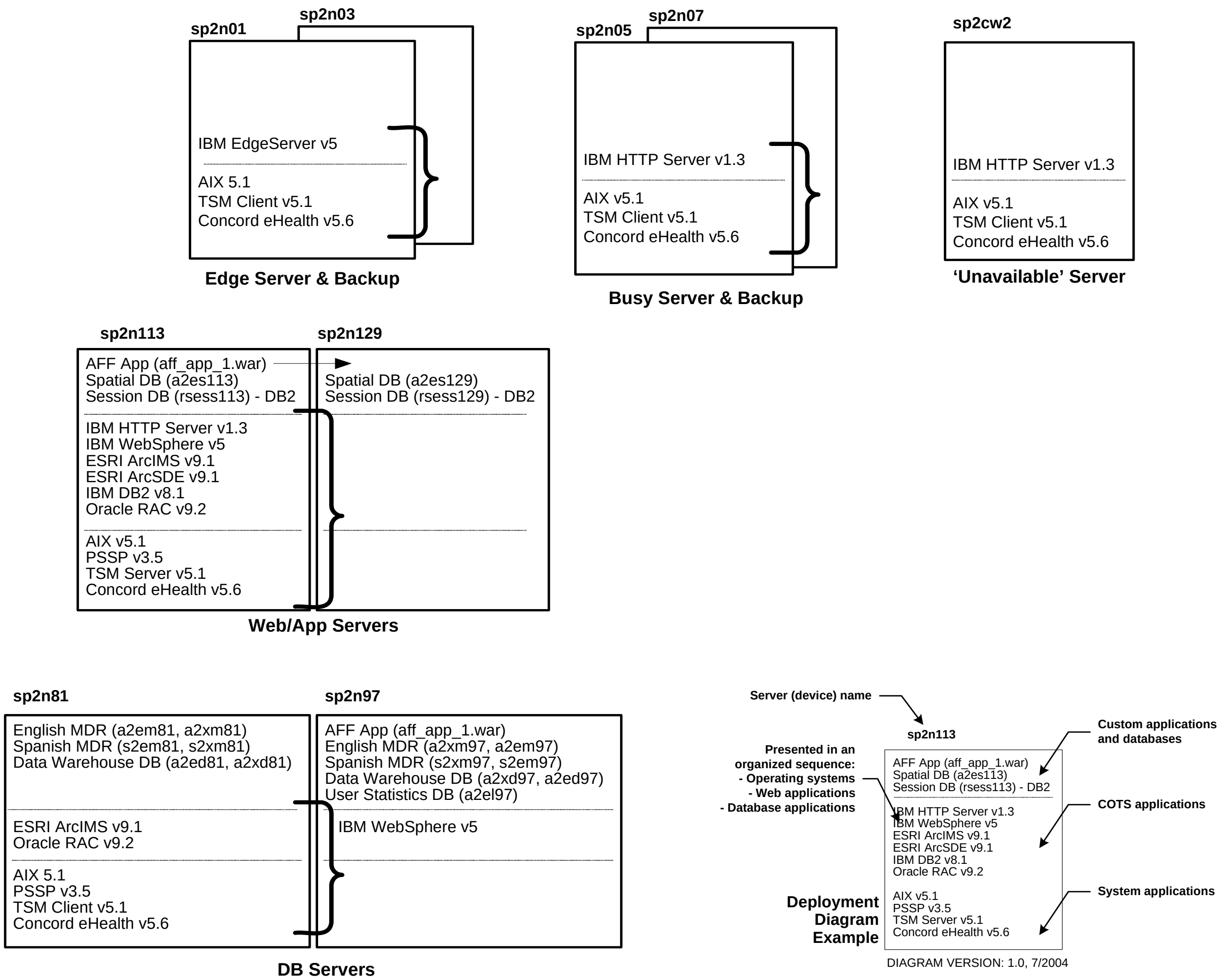

Figure 6: AFF Software Deployment - Production

The Internal Review and development environments are variations of the above figure.

3.2.4. AFF Operations Scenarios

Tables and maps are the census data products disseminated by AFF. These data products are built from well-defined queries against the pre-tabulated data stored in the Data Warehouse DB. The query parameters are specified interactively by the user following the user interface navigation path in the AFF application and controlled by the information in the Metadata Repository databases. The results of the query are formatted according to pre-defined templates defined in the AFF application's presentation layer and derived from information in the metadata repository. In general, all interactions between the user and AFF are a series of transactions that either gather query parameters or request and present query results. See the following four sample scenarios for examples of the transaction flows.

3.2.4.1. Search-by-Address Scenario

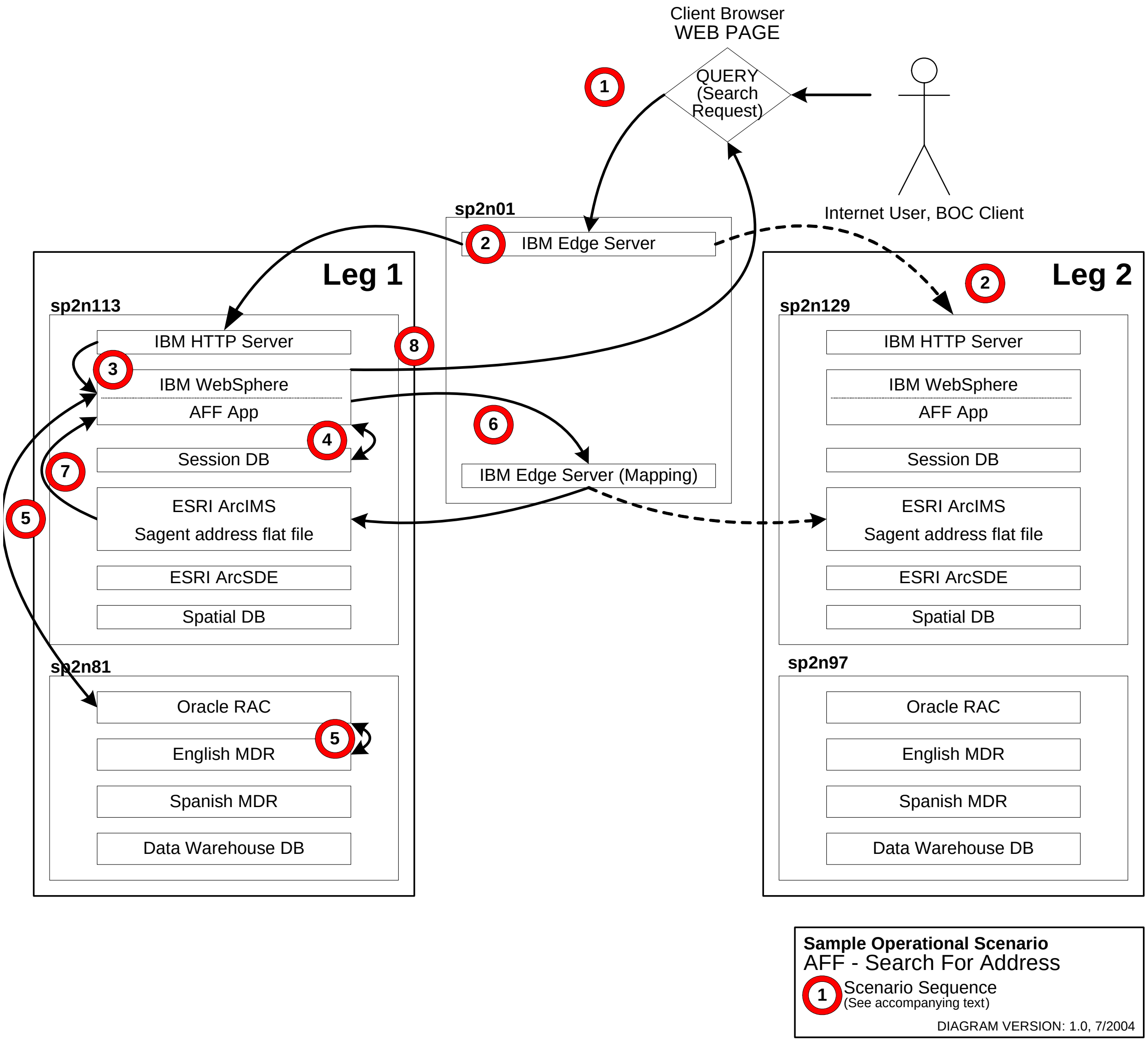

Figure 7: AFF Operational Scenario - Search-by-address

1. An Internet User creates a search-by-address (SBA) query through the AFF SBA web page. When the user clicks the Search button, the Client Browser sends an HTTP request with the SBA query parameters to the URL associated with the SBA's named servlet.
2. The query is received by the IBM Edge Server and dispatched to an IBM HTTP Server based on the system's load-balance conditions.
3. The IBM HTTP Server detects that the request is a named servlet (a dynamic request) and forwards the query to IBM WebSphere/AFF Application (WebSphere/AFF).
4. WebSphere/AFF retrieves the session ID from the request and updates the session in the Session DB with the latest query parameters; the session is created if it doesn't exist.
5. Based upon the user's query parameters, WebSphere/AFF gathers any needed metadata associated with the query from the MDR using the Oracle RAC.

6. WebSphere/AFF creates a SBA query from the user's data and the metadata and sends the SBA query to ArcIMS via the IBM Edge Server (Mapping) for secondary load-balancing between mapping subsystems; the IBM Edge Server dispatches the SBA query to an available ArcIMS server on either production leg.
7. ArcIMS locates address data from the data file supplied by Sagent and returns the query results to WebSphere/AFF.
8. WebSphere/AFF formats the query results into an HTML page and returns the completed page to the Client Browser using an HTTP Response.

This scenario shows the AFF multi-leg capacity. The IBM Edge Server and IBM Edge Server (Mapping) may deploy the query to one of two legs (dotted-line alternatives for steps #2 and #6 above), depending on current system conditions. The remaining sample scenarios will just show one leg for clarity.

3.2.4.2. View Quick Table Scenario

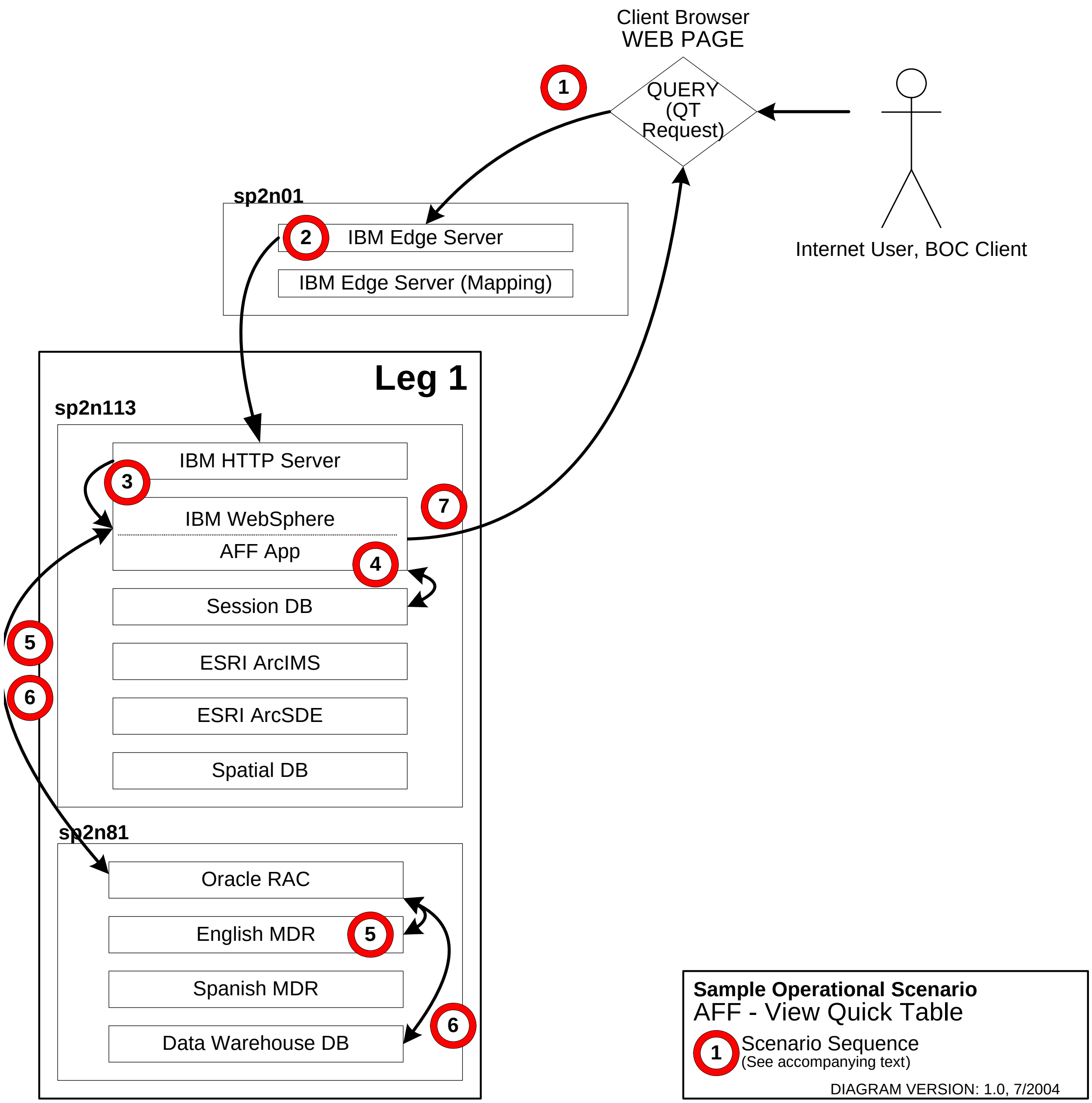

Figure 8: AFF Operational Scenario - View Quick Table

1. An Internet User creates a quick table (QT) query through the AFF QT web pages. When the user clicks the Show Results button, the Client Browser sends an HTTP request with the QT query parameters to the URL associated with the selected QT result page.
2. The query is received by the IBM Edge Server and dispatched to an IBM HTTP Server based on system's load-balance conditions.
3. The IBM HTTP Server detects that the request is a named servlet (a dynamic request) and forwards the query to WebSphere/AFF.

4. WebSphere/AFF retrieves the session ID from the request and updates the session in the Session DB with the latest query parameters; the session is created if it doesn't exist.
5. Based upon the user's query parameters, WebSphere/AFF gathers any needed metadata associated with the query from the MDR, and builds a database query from the user's query parameters and the metadata.
6. WebSphere/AFF uses the query built in the previous step to request tabular data from the Data Warehouse using the Oracle RAC.
7. WebSphere/AFF formats the query results into an HTML page and returns the completed page to the client browser using an HTTP Response.

3.2.4.3. View Map Scenario (Spanish)

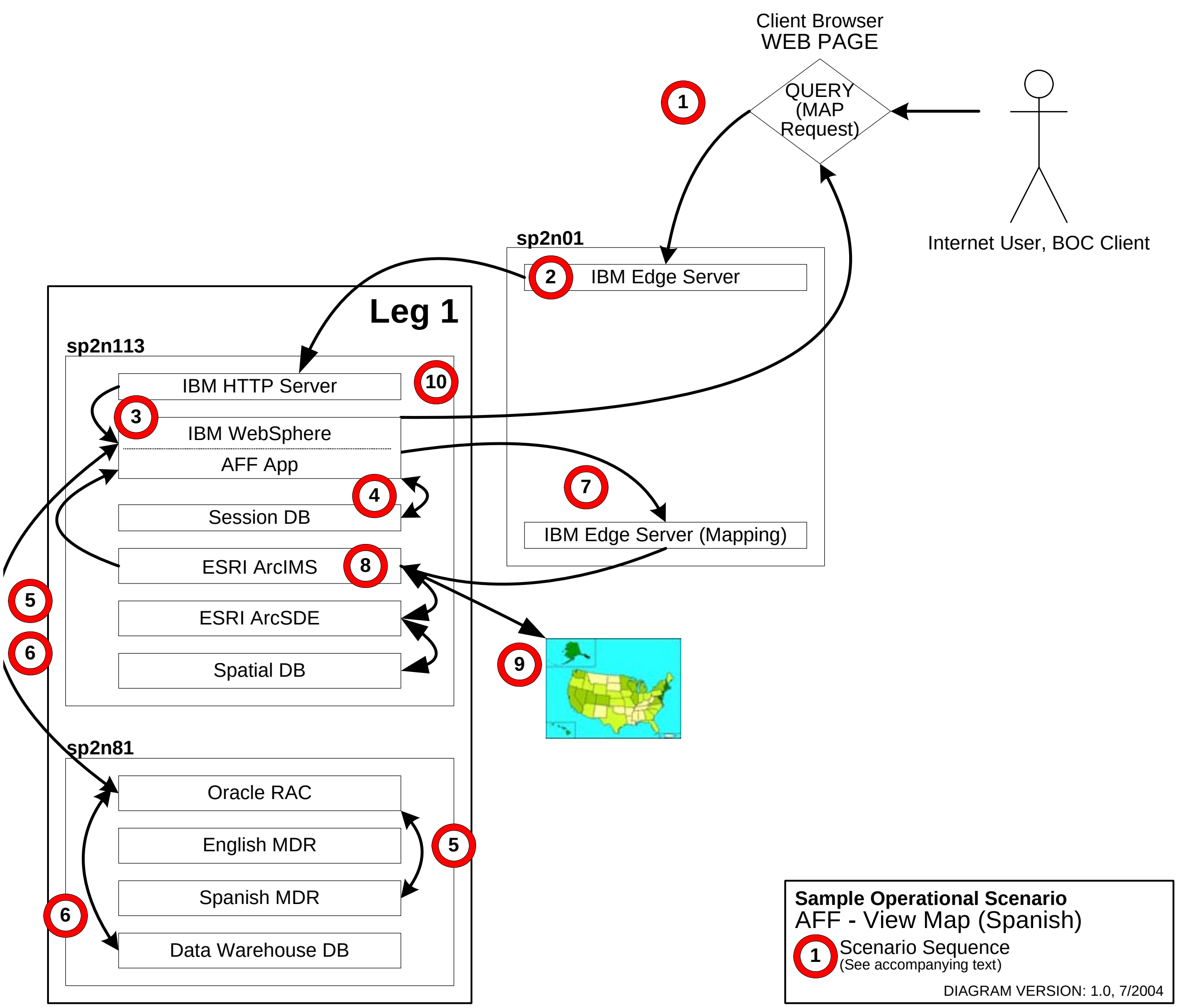

Figure 9: AFF Operational Scenario - View Map (Spanish)

1. An Internet User creates a thematic map query through a Spanish language AFF thematic map (TM) request web page. When the user clicks the Show Results button, the Client Browser sends an HTTP request with the TM query parameters to the URL associated with the TM's named servlet.

2. The query is received by the IBM Edge Server and dispatched to an IBM HTTP Server based on system's load-balance conditions.

3. The IBM HTTP Server detects that the request is a named servlet (a dynamic request) and forwards the query to IBM WebSphere/AFF Application (WebSphere/AFF).

4. WebSphere/AFF retrieves the session ID from the request and updates the session in the Session DB with the latest query parameters; the session is created if it doesn't exist.

5. Based upon the user's query parameters, WebSphere/AFF gathers any needed metadata associated with the query from the Spanish language MDR using the Oracle RAC.

6. WebSphere/AFF builds a query using the user's query parameters and the supporting metadata to request the data values associated with the theme and geography from the tabular data in the Data Warehouse using the Oracle RAC.

7. WebSphere/AFF creates a TM request from the user's data, the metadata, and the data values retrieved from the Data Warehouse, and sends the TM request to ArcIMS via the IBM Edge Server (Mapping) for secondary load-balancing between mapping subsystems; the IBM Edge Server dispatches the SBA query to an available ArcIMS server on either production leg.

8. ArcIMS retrieves the spatial data for the map image from the Spatial DB using the ESRI ArcSDE server.

9. ArcIMS creates the specified thematic map based upon the parameters and data values in the map request and the spatial data retrieved from the Spatial DB. The thematic map is written as a GIF file to a shared file system accessible by the IBM HTTP Server; ArcIMS returns the name of the GIF file to WebSphere/AFF.

10. WebSphere/AFF formats the map result page embedding the map file name as an HTML IMG tag and returns the completed page to the Client Browser using an HTTP Response.

11. While processing the response page, the Client Browser retrieves the map image as a static file from the IBM HTTP Server (steps A, B, C and D in the following figure):

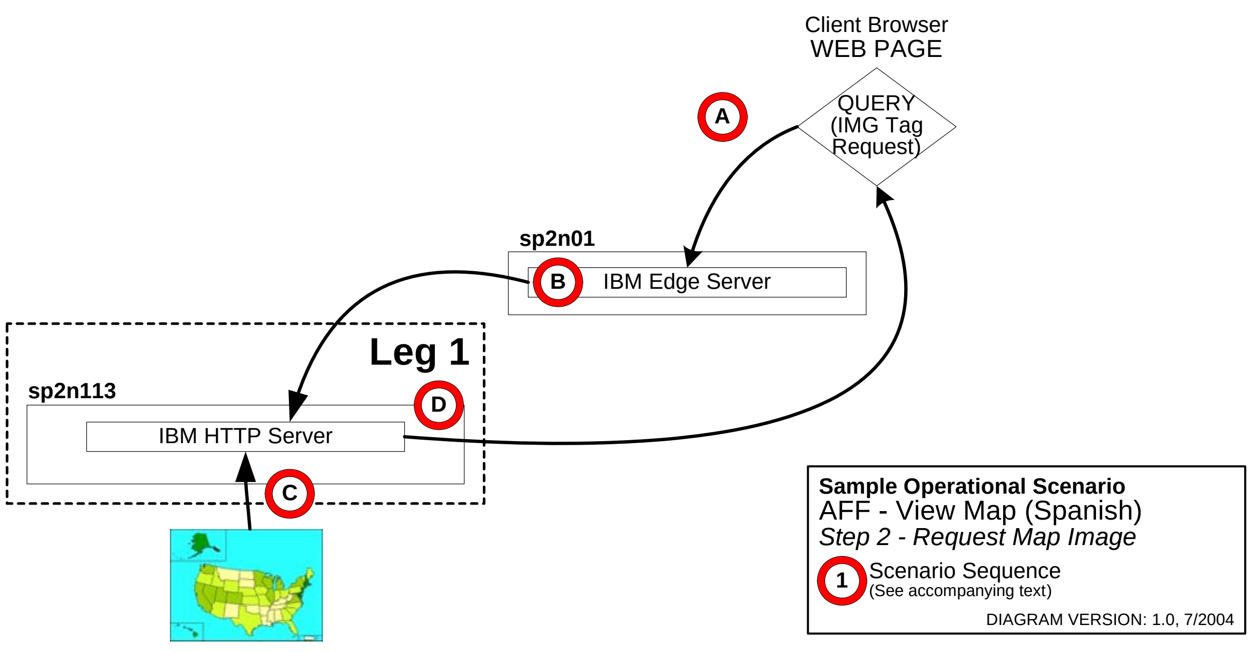

Figure 10: AFF Operational Scenario - View Map (Spanish) - Request Map Image

3.2.4.4. View Topic Page Scenario

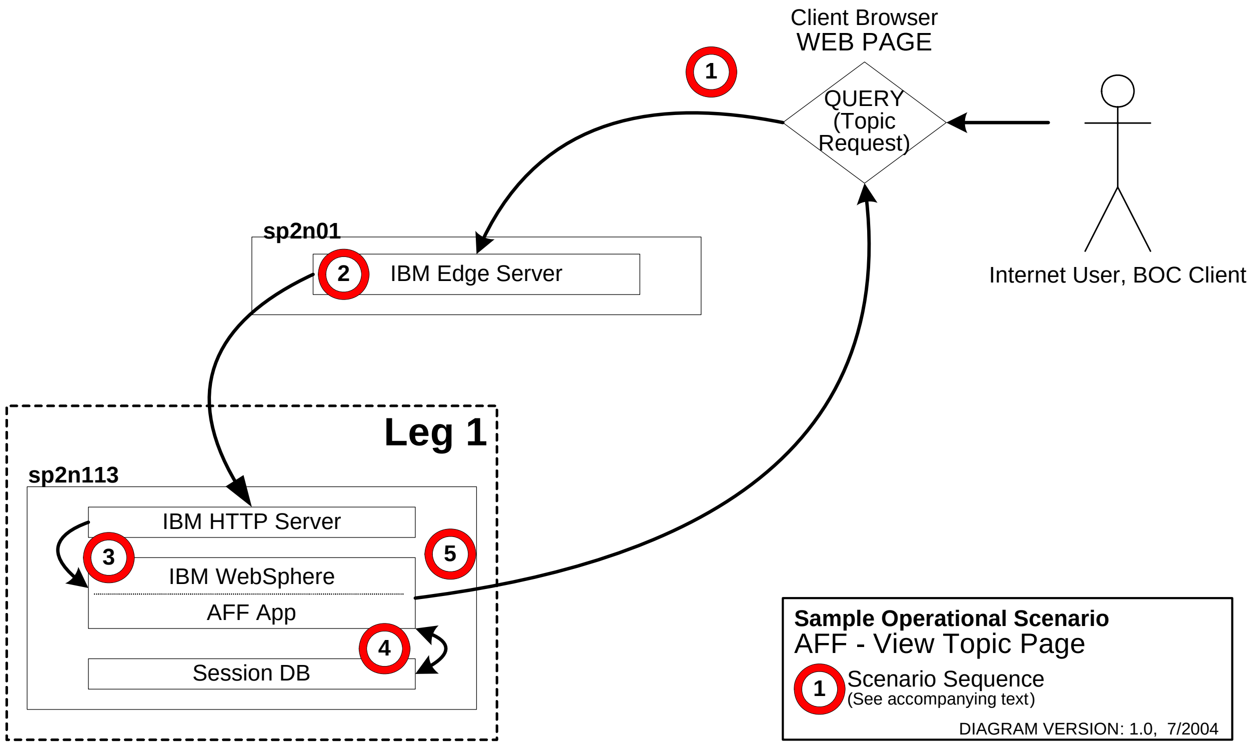

Figure 11: AFF Operational Scenario - View Topic Page

1. An Internet User creates a topic page request through a link on an AFF web page. When the user clicks the link, the Client Browser sends an HTTP request with the topic request identifier in the URL to a URL associated with an executable JSP.
2. The query is received by the IBM Edge Server and dispatched to an IBM HTTP Server based on system's load-balance conditions.
3. The IBM HTTP Server detects that the request is a named JSP (a dynamic request) and forwards the query to IBM WebSphere/AFF Application (WebSphere/AFF).
4. WebSphere/AFF retrieves the session ID from the request and updates the session in the Session DB with the latest query parameters; the session is created if it doesn't exist.
5. WebSphere/AFF invokes the specified JSP and the JSP dynamically creates a HTML page on the topic page specified by the request parameters by including a static content page inside a common topic page framework. The completed page is returned to the Client Browser using an HTTP Response.

3.2.5. AFF Hardware Architecture

Please reference the Attachment document for more details on the AFF Hardware Architecture.

3.2.6. AFF Network Architecture

Please reference the Attachment document for more details on the AFF Network Architecture.

4. ADVANCED QUERY (AQ) ARCHITECTURE

The Advanced Query (AQ) architecture is described in this section using Logical and Physical views. The Logical view includes:

- Design Influences
- Application Architecture Overview

The Physical view includes:

- Design Influences
- Physical Overview
- Software Deployment
- Operational Scenarios
- Hardware Architecture
- Network Architecture

4.1. AQ Logical Architecture

The Advanced Query System produces on-line tabulations from the Census 2000 microdata files on the Internet. The system gives users the ability to construct tabulations from the Hundred Percent Data File (HDF) and the Sample Edited Data File (SEDF) with confidentiality filters based on rules for electronic disclosure limitation developed by the Census Bureau.

From a business perspective, the most significant feature of the Advanced Query system is that is allows access to Census data in a way that was never possible before. Properly authorized users can use the Internet to perform ad-hoc queries against microdata from the Decennial Census to form cross-tabulation results not in the published set of Decennial summary files. Interestingly, these ad-hoc queries are still subject to automatic disclosure limitations that examine each query on case-by-case basis for disclosure problems. Only those queries that pass all of the pre-defined Census disclosure rules are returned to the user.

Disclosure limitations are implemented through a pair of AQ filters: the Query Filter and the Results Filter. The Query Filter is designed to detect those queries that will not pass disclosure limitations before they are submitted for execution. This saves database resources and optimizes the user experience by quickly identifying cross-tabulation options that would result in data the user is not entitled to view. The Results Filter works on data that has already passed the query filter's criteria. The results filter is designed to execute the selected cross-tabulation against the data warehouse, and then remove any geography whose results fall below a specified mean, median, or sparsity threshold.

The geographic coverage for HDF and SEDF is the same as the Decennial 2000 Summary File products SF1 and SF3, respectively, although the smallest geography areas for HDF and SEDF are whole block groups and whole census tracts, respectively.

4.1.1. Design Influences

Like AFF, the number of possible navigation paths and result pages in AQ is extremely large. Also like AFF, a design decision was made to make AQ a dynamic database-driven system.

Based on the requirements to support ad-hoc queries against the HDF and SEDF microdata, the need to house HDF and SEDF geography hierarchies, the large number of universes and demographic categories combinations, and the need to implement BOC-defined confidentiality filters, OLAP-based tools were evaluated to implement Advanced Query. Relational-based OLAP (ROLAP) was identified as the best fit for the Census requirements. A multi-dimensional OLAP (MOLAP) database approach (like

Cognos) that pre-builds all possible cross-tabulations was determined to be infeasible, given the extremely large number of possible demographic variables and geographic areas in the system.

The leading ROLAP COTS software on the market today for Internet dissemination is MicroStrategy. MicroStrategy was chosen for its support for ad-hoc queries against a larger number of categories, and its support for the Confidentiality Filters – specifically to automatically insert the Query Filters and the Result Filters into the user's interaction with the data.

In particular, the MicroStrategy solution provides:

- Browser-based client support
- High performance on large databases
- Firewalls between the web application and the databases
- The flexibility to add Title 13 filters on cross-tabulation results
- GUI customizations to present easy to read web pages based on the BOC web site style standards
- Ability for users to interactively define tabulations and optionally queue them for background processing
- Ability for users to save and manage tabulation reports

4.1.2. AQ Logical Application Architecture

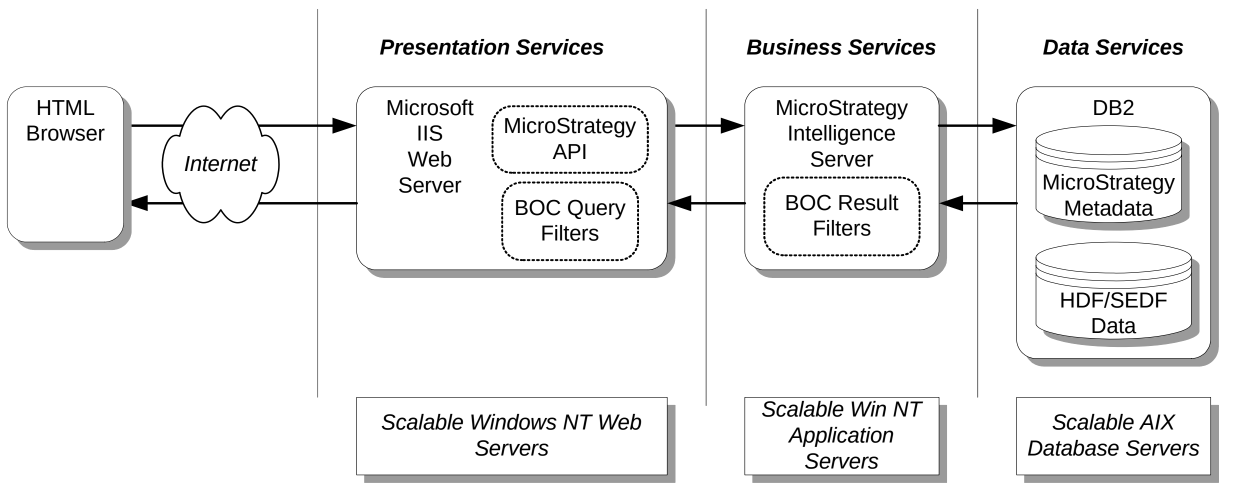

Figure 12: AQ Logical Architecture Overview

Users create and execute queries interactively within an HTML Browser via a series of requests to AQ. These requests are received by the Microsoft IIS Web Server and forwarded to the MicroStrategy Intelligence Server for processing. The query filters are implemented in the Microsoft IIS Web Server. MicroStrategy, an OLAP tool, uses the Metadata stored in a DB2 UDB database to build queries to tabulate detail HDF or SEDF data stored in another DB2 UDB database. The resulting table data is passed through statistical result filters in order to ensure that no sparse tabulations are released. The result table data, formatted by the MicroStrategy Intelligence Server as either a table or chart, is returned to the user by the Microsoft IIS Web Server.

Access is controlled by passwords and physical network segregation, filtering is controlled by custom application coding.

In order to create a report using the Advanced Query system, users execute the following steps:

- External users (authorized members of the public) access Advanced Query using the Internet at http://advancedquery.census.gov/; Internal users (authorized BOC users) access Advanced Query using the Intranet at http://t3inws1.dads.census.gov/
- Selects either the HDF or SEDF microdata to perform a tabulation.
- Select a single geographic type (summary level) for the report. There can only be one type of geography on each report. For example, a report can contain only states *or* counties, not state *and* counties.
- Select the geographies of interest. For example, select all the counties in the United States. The population of the smallest geographic area is tracked and controls the subsequent choice of demographic characteristics.
- Choose the universe (what to count). The main universe categories are "All People", "People in Households", and "Households and Housing Units", each with many subcategories. A report can count only one type (universe).
- Select the characteristics on which your tabulation will be based. You can choose up to three characteristics for the report. The population size of the smallest geographic area in the user's selection controls the demographic characteristics available.
- Steps 2-5 are subject to Query Filters that protect against indirect disclosure of confidential information that might occur by subtraction of one table from another when multiple tables are examined. The Query Filters are enforced *before* the tabulation is submitted.
- The user executes the tabulation request, or can optionally queue the request and retrieve the results at a later time.
- The requested tabulation is computed. Before returning the results to external users, the Results Filter is applied. These tests check for sparse tabulations using a mean, median, and sparsity test. Internal users are not subject to the result filters.
- Geographies that pass the Result Filters are displayed in an OLAP report view that allows the user to sort, pivot and chart the results. Geographies that fail the Result Filters are withheld and documented on-line in a separate nondisclosure report. To see data, at least one geography must pass the filters. The Result Filers are enabled only for external users of the Internet-based AQ external system.

The AQ system also allows users to save, edit, and queue tabulation requests as part of their login session. Like AFF, AQ collects application usage statistics in order to create user activity reports. Likewise, to understand how this same activity affects AQ itself, AQ collects system performance information in order to create system activity reports.

To understand how its users perceive AQ, AQ provides a mechanism for feedback collection that allows users to request help, report defects and suggest enhancements.

For external users, the web server and application/data servers are separated by a firewall for extra security. No user (internal or external) can directly browse the underlying microdata.

4.1.3. AQ Logical Data Structure

The AQ data is stored in a metadata repository managed by MicroStrategy and a Data Warehouse that contains the HDF and SEDF databases. The Data Warehouse is organized using star schema structures for optimal performance.

The HDF and SEDF databases are associated with different data universes and do not share data; MicroStrategy accesses these databases independently. Users are prompted to select the universe (HDF or SEDF data) as a part of their cross-tabulation definition.

For HDF, "profiles" were created to reduce the number of demographic records. A profile is a unique combination of selected demographic characteristics regardless of the geography. A profile is created independently of the record's geographic identification.

Three profile records were created for both Housing and Person records. The person profile table condenses the 280 million person records from the source down to 617,059 distinct profile records. The housing profile table condenses the 10 million housing records from the source to 74,400 distinct profile records. There are 24 attributes in the person external profile and 10 in the housing external profile. The HDF profiles are an implementation approach that saves considerable disk space but have no functional effect on the end user. Internal profile tables (available only to internal AQ users) were created with additional fields. Operational Profiles, also only available to internal AQ users, were also created with fields about how the data was collected.

For SEDF, the 1/6 reduction in the number of microdata records made the profile technique unnecessary.

4.2. AQ Physical Architecture

The AQ system provides on-line tabulations of the low-level census data found in the HDF and SEDF into Title 13 tables safe for dissemination. The AQ server hardware and network are located in the Bowie Computer Center Mod 2 Cage and Mod 6 Cage. The AQ developer and operator workstations are located in Suitland.

4.2.1. Design Influences

AQ is a web-enabled production system that allows user-defined cross-tabulations of low-level census data. There are two classes of users:

- **Internet (external) Users** must have tabulation results that pass Title 13 disclosure restrictions; these users can only access AQ using the internet.
- **BOC Intranet (internal) Users** must have the ability to relax the Title 13 disclosure restrictions; these users can only access AQ using the BOC intranet.

Given that AQ requires in-depth understanding of BOC data and access is restricted, AQ must only support 300 concurrent users. The relatively low-level of data and traffic on AQ means that only a single network is required to support AQ unlike DADS and AFF.

The choice of the MicroStrategy COTS software package choice drove other decisions based upon the platforms and technologies supported by MicroStrategy.

At the time of initial implementation, Windows-based servers were required for the MicroStrategy Web Analyst and Intelligence Server applications. In addition, the MicroStrategy solution requires a front-end web server; the best choice at the time was the Microsoft Internet Information Services (IIS) also running on a Windows-based Server.

Otherwise, standard DADS AIX platforms were used for database and metadata servers. Within DADS, this mixture of Windows and AIX servers is unique to AQ.

IBM DB2 was selected as the back-end database based upon MicroStrategy's high performance integration with DB2 UDB running on scalable AIX database servers.

Components of the AFF system architecture for availability and load balancing are reused for the AQ system in order to provide a robust application. These architectural components include redundant web and application servers and the use of edge servers, busy servers and unavailable servers.

4.2.2. AQ Physical Overview

The AQ hardware consists of multiple servers as shown in **Figure 13:** AQ Physical Overview.

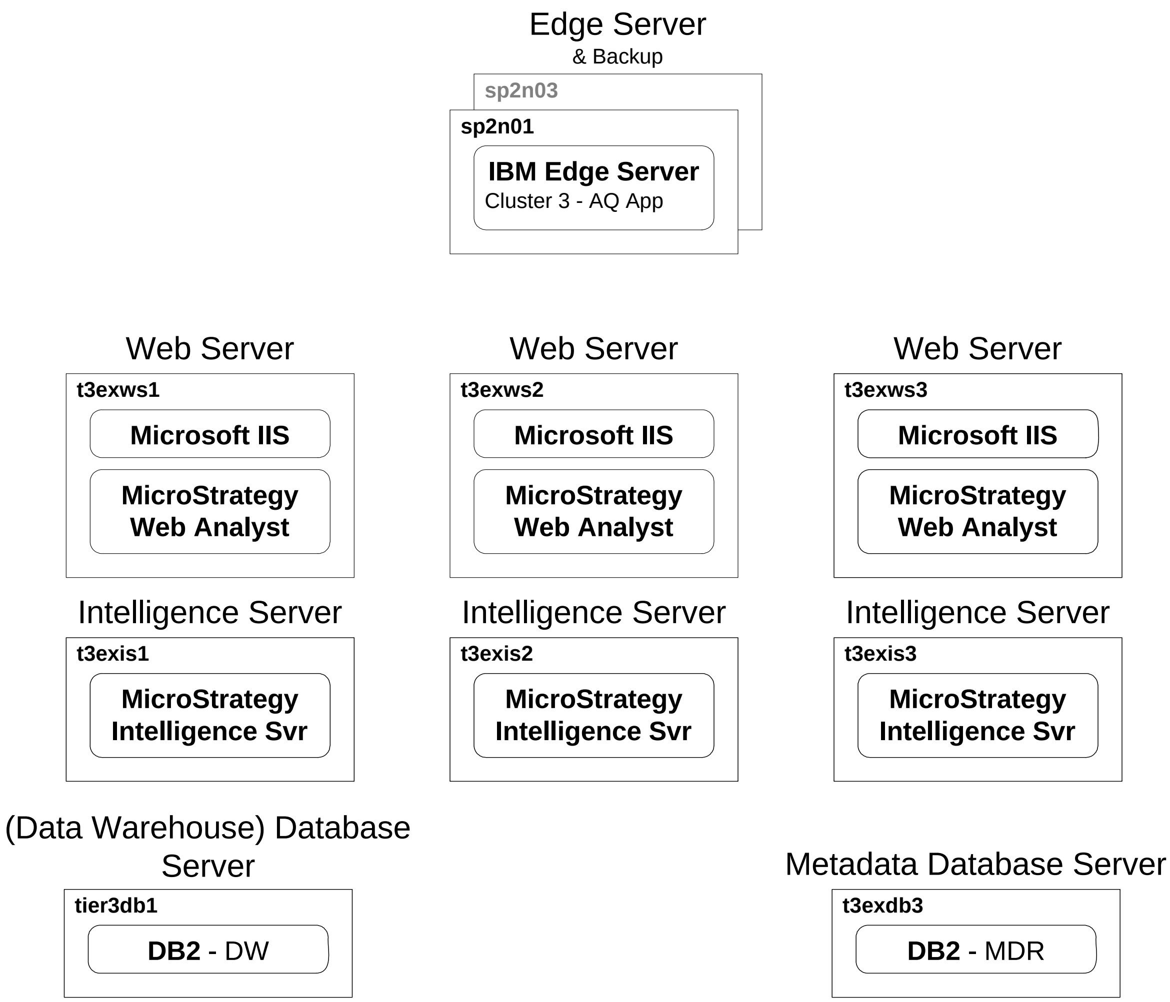

Figure 13: AQ Physical Overview

The **Edge Servers**, shared with AFF, are used to distribute incoming requests between the **Web Servers**.

Multiple **Web Servers** and **Intelligence Servers** are used to distribute the processing in order to meet performance requirements and security concerns.

The AQ system has two database servers, **tier3db1**, the **(Data Warehouse) Database Server**, and **t3exdb3**, the **Metadata Database Server**. Both database servers use the **DB2 UDB** database system.

The **Web Servers**, using a Microsoft Windows Server operating system, run the **Microsoft Internet Information Server** along with the **MicroStrategy Web Analyst** to receive and process user requests and to build and return responses. The **Intelligence Servers**, using a Microsoft Windows Server operation system, run the **MicroStrategy Intelligence Server** software to handle data requests between the **Web Servers** to the database servers.

A secure network and interface exists between the **Web Servers** and the **Intelligence Servers** in order to isolate and secure the data.

In order to handle the data volumes and processing requirements, the **Web Servers** and **Intelligence Servers** are multiprocessor systems. If needed, the architecture allows additional servers to be added to the AQ system in order to meet the workload.

To meet system performance requirements, the **Database Server** is a multiprocessor AIX system. To ensure data availability, the **(Data Warehouse) Database Server** is attached to a SSA Disk Farm with RAID storage and an automated tape backup system.

4.2.3. AQ Software Deployment

The AQ application is a set of customized MicroStrategy Business Intelligence Platform reports that conform to the BOC web site guidelines. In addition to normal configuration, AQ required minor customization of the MicroStrategy software in order to implement the result filters needed for AQ to meet Title 13 disclosure restrictions. Otherwise, AQ is a standard and configuration installation of the MicroStrategy Business Intelligence Platform Commercial Off-The-Shelf (COTS) software. The MicroStrategy product required the use of Windows-based servers at the time of initial implementation; this fact drove the decision to use Microsoft Active Server Pages (ASP) and Microsoft Component Object Model (COM) software architectures in the configuration, customization and enhancements to the MicroStrategy base software.

AQ software is divided into three categories:

- **Custom** – Applications and databases written to support a system function or enhancement
- **COTS** – Vendor-produced software, also known as Commercial Off-The-Shelf (COTS) software
- **Server** – Applications supporting the operation and administration of the servers and network

The following table is a high-level matrix of the AQ software and databases grouped by the above categories:

	Software	Description
Custom	SRD.exe	Statistical Research Div. Program storing user reports generated on AQ into Excel spreadsheet.
	Census.dll	loads XML navigation menus into memory
	SEDFDataFiles	VB application that produces XML navigation menus for the SEDF project
	HDFDataFiles	VB application that produces XML navigation menus for the HDF project
	BOCLMedian	Visual C++ custom MicroStrategy function to calculate Linear Median
	BOCPMedian	Visual C++ custom MicroStrategy function to calculate Pareto Median
	Active Server Pages	Out of the box MicroStrategy ASP pages customized for Census look and feel and additional functionality
	HDF DB	Formatted HDF data
	HDF Staging DB	Raw HDF data
	SEDF DB	Formatted SEDF data
	SEDF Staging DB	Raw SEDF data
COTS	Microsoft IIS v5	Patch mgmt. for networks with more than 25 PCs.
	MicroStrategy (MSTR) Web Analyst v7 – customized	MicroStrategy Business Intelligence Platform v7 – core software to AQ design and operation
	MSTR Intelligence Server v7	
	MSTR Desktop Analyst v7	
	MSTR Administration v7	
	IBM DB2 UDB v8	Database operating system – DBMS supporting HDF and SEDF databases
	IBM DB2 UDB Client v8.1	
	IBM DB2 UDB for AIX v8	
	Oracle Client v9.2	Responds to user selection of ‘Feedback’ request and points to shared Oracle DB
System	MS Windows 2000	Server operating system
	MS Internet Information Services (IIS)	Web server software
	TSM Server v5.1	Tape back-up system
	TSM Client v5.1	Tape back-up system
	Concord eHealth v5.6	Web-based performance monitoring tool

Table 4: AQ Software

The following figure shows the distribution of the above software on the AQ production hardware used for external Internet users. In order to meet the requirements for availability, self-healing and on-demand, the AQ production system has redundant independent Web Servers and Intelligence Servers.

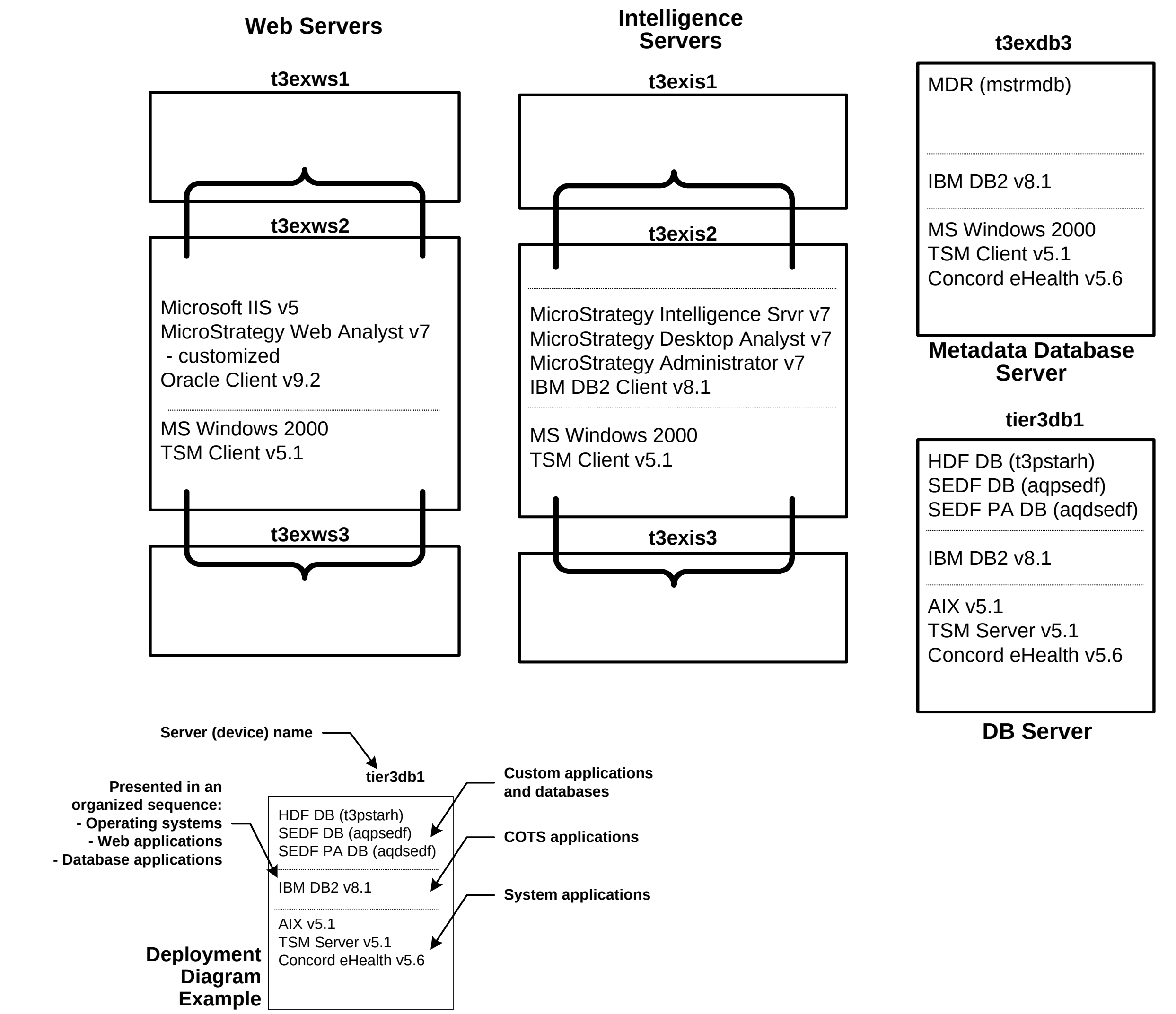

Figure 14: AQ Software Deployment - Production

The production and PA Data Warehouse is physically located on tier3db1 within the production DMZ.

The following figure shows the distribution of the above software on the AQ hardware used for internal BOC intranet users and development.

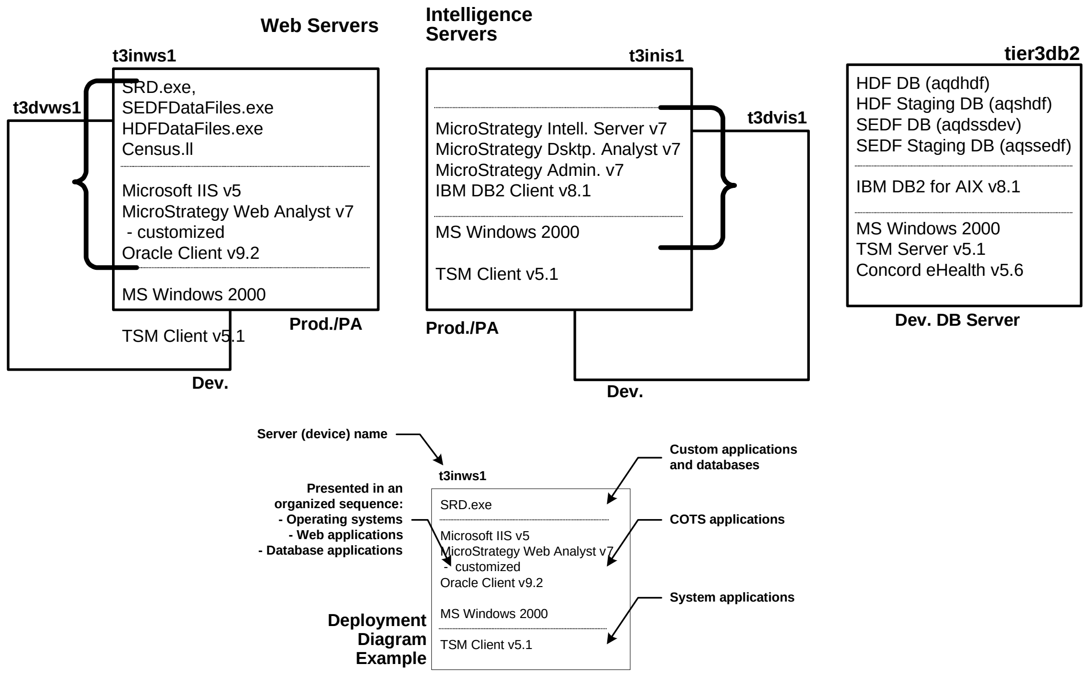

Figure 15: AQ Software Deployment - Intranet Production

The development Data Warehouse is physically located on tier3db2.

The above figures do not include the various Windows workstations used by developers and testers to develop and test AQ. These developers use the MicroStrategy desktop client.

4.2.4. AQ Operations Scenarios

Tables and charts are the census data products disseminated via AQ. These data products are built from user-defined cross tabulations against the SEDF and HDF data. Access to AQ is restricted via passwords to Internet and Intranet users who have been registered for secured access to AQ. Query parameters are specified interactively by the user following pre-defined scripts defined in the AQ application and controlled by the information in the Metadata. The results of the query are formatted according to pre-defined DHTML templates defined in the AQ application's presentation layer. In general, all interactions between the user and AQ are a series of transactions that either gather query parameters or request and present query results. See the following two sample scenarios for examples of the transaction flows.

4.2.4.1. Internet Query Scenario

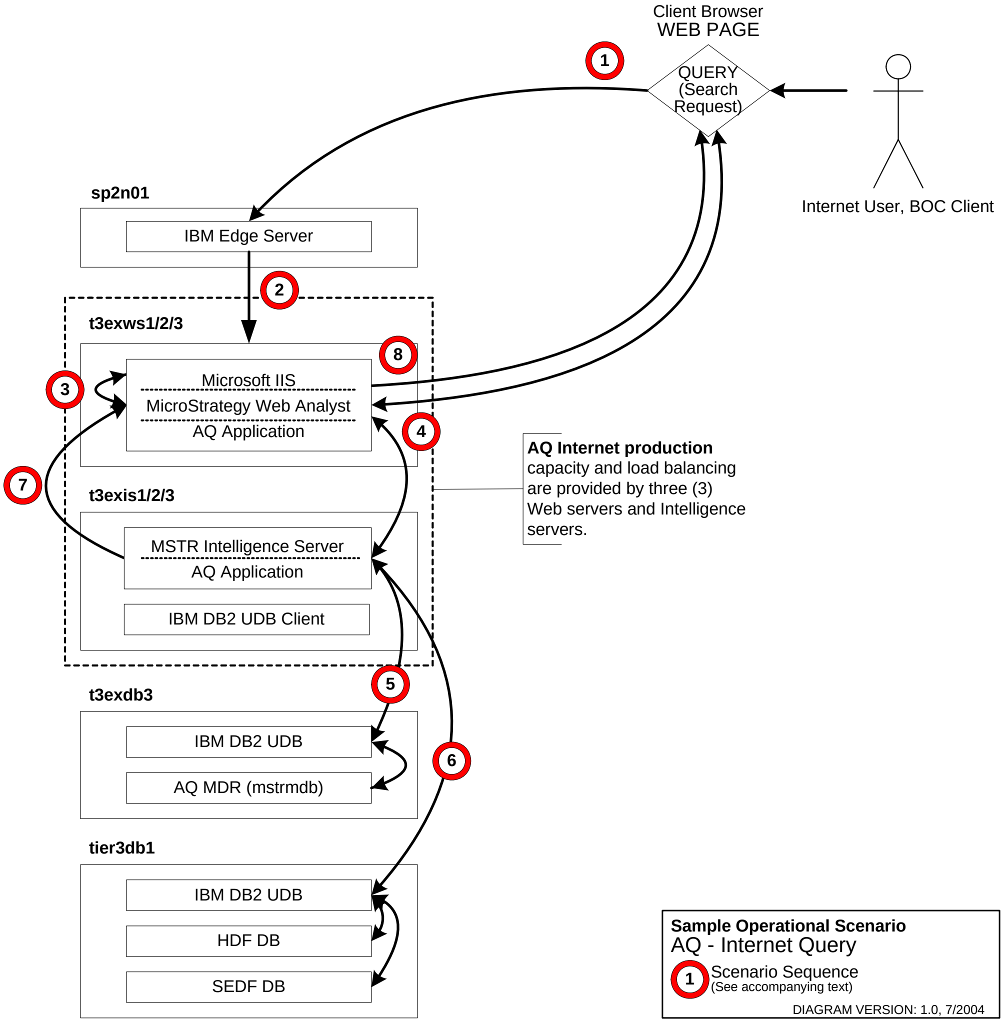

Figure 16: AQ Operations Scenario - Internet Query

1. An Internet User logins in to AQ using an AQ user ID and password. The login sends an HTTP request with a MicroStrategy Web Analyst URL.
2. The login is received by the IBM Edge Server (shared with AFF) and dispatched to a Microsoft Internet Information Services (IIS) server based on the system's load-balance conditions.
3. Microsoft IIS detects that the URL is associated with MicroStrategy Web Analyst and forwards the logon request to MicroStrategy Web Analyst for processing. Web Analyst:
   - Authenticates the User ID and password
   - Creates and manages the user's session
   - Works with the MicroStrategy Intelligence Server to resolve user queries
4. The Internet User interactively defines cross-tabulations in a series of HTTP requests and responses using Web Analyst where the user navigation and selections associated with the cross-tabulation are defined by the AQ Application and the MicroStrategy Metadata.
   - Query filters are applied iteratively during the selection process

5. Access to the MicroStrategy Metadata is provided by the MicroStrategy Intelligence Server using IBM DB2 UDB.
6. When the user submits a cross-tabulation for processing, the Web Analyst passes the cross-tabulation definition to the Intelligence Server. The Intelligence Server parses the cross-tabulation request into a series of SQL queries against the HDF and SEDF databases.
   - Data privacy (Title 13) statistical result filtering is automatically applied.
7. The results of the database queries are processed by the Intelligence Server and passed to the Web Analyst.
   - Data privacy (Title 13) statistical result filtering is automatically applied.
8. Web Analyst formats the results into a DHTML page and returns the page to the Client Browser as an HTTP response.

4.2.4.2. BOC Intranet Query Scenario

The operational scenarios for queries through BOC Intranet access and Internet access are the same with the following exceptions:

- Different IDs and passwords are required to access AQ from the BOC Intranet versus the internet
- Users with intranet IDs must access the system using the BOC intranet
- The level of data privacy filtering (Title 13) can be adjusted by BOC intranet users

4.2.5. AQ Hardware Architecture

Please reference the Attachment document for more details on the AQ Hardware Architecture.

4.2.6. AQ Network Architecture

Please reference the Attachment document for more details on the AQ Network Architecture.

5. DATA PRODUCT PRODUCTION (DPP) ARCHITECTURE

The Data Product Production (DPP) architecture is described in this section using Logical and Physical views. The Logical view includes:

- Design Influences
- Application Architecture Overview

The Physical view includes:

- Design Influences
- Physical Overview
- Software Deployment
- Operational Scenarios
- Hardware Architecture
- Network Architecture

5.1. DPP Logical Architecture

The DPP system is a general-purpose data product production system. It produces Census data products (namely, Summary Files) based on product specifications defined by POP/HHES and microdata files generated from the 2000 Decennial Census. It is a programmed batch system integrated with commercial off-the-shelf (COTS) products, including SAS, SuperSTAR, and the UNIX Korn shell.

DPP was designed to tabulate the Census 2000 Hundred Percent Data File (HDF) and Sample Edited Data File (SEDF) into summary files for dissemination by AFF and ACSD. Expert users also download the summary files directly from the public Census FTP site to perform off-line analysis. Summary Files are a standard text-based file format used to distribute Decennial Census data.

The DPP system was used to produce the official Decennial Census 2000 tabulated results, including Summary File P.L. 94-171, used for Congressional Redistricting. It is also used to produce special tabulations – most recently, a series of School District tabulations for the National Center for Education Statistics (NCES). It will also be used to tabulate Census 2000 data along the new 109th Congressional District boundaries.

5.1.1. System Inputs

There are three major types of input files to the DPP system: **Microdata files**, **DPP Geography Files**, and **Independent Tabulation Files**. Microdata files are the files that contain the characteristics of people and housing units from the census operations. The microdata records are identical to the set used by the HDF and SEDF Advanced Query application. DPP Geography Files contain the information DPP uses to tabulate each housing unit and person into the appropriate summary levels. Independent Tabulation Files (also called Analyzer Files) contain the results of independent tabulations used to check against the DPP results.

The other significant DPP input is the **product specification** from POP/HHES. This document defines the nature of the data product, specifically its geographic coverage including geographic components, required matrix table definitions, derived measures, thresholding, rounding, and disclosure avoidance requirements.

5.1.2. DPP Generated Inputs

In addition to these external inputs from **DSCMO** (microdata files and analyzer files), **POP/HHES** (product specification), and **Geography Division** (geography files), the DPP team also prepares a set of inputs before tabulation begins –

- **Driver files:** A set of driver files – a set of parameter text files that define the data product to the DPP system.
- **TXDs:** Using the SuperCROSS client, subject matter experts compose table definitions consistent with the product specification. Generally, there's one table per product matrix table definition. Table definitions are saved as **textual table definition** (**TXD**) files. DPP calls these the "template TXDs" because during runtime, the templates are merged with other recodes to produce the actual TXD used for production tabulation.
- **Characteristic Iterations:** Subject matter experts compose the race, ancestry, ethnicity, or tribal recodes used in iterated products like Summary File 2, Summary File 4, American Indian and Alaska Native (AIAN), and the various School District special tabulations. These files are called characteristic iteration (CI) recodes and are TXD fragments merged into the template TXDs during production tabulation.

Two critical intermediate outputs produced during production are the following:

- **Geographic Recodes:** Based on the geography inputs, a set of geographic recodes is programmatically created using the ProcessGeo script before tabulation. Geographic recodes convey to the tabulation engine which geographies to tabulate. Generally, there's one geographic recode file per state, although many other configurations are possible (discussed below).
- **SXV4 Detail Databases:** Based on the microdata files, a set of databases in the native SXV4 SuperCROSS format are created using the SuperCHANNEL database build utility. SXV4s are highly optimized, proprietary databases that work with the SuperSERVER tabulation engine on AIX.

5.1.3. System Outputs

There is only one major type of output file from the DPP system. It is the **Summary File**. Summary File is a logical designation; in fact, a Summary File is a collection of two or more physical files. The structure of the Summary File is defined by DPP in the "Details of the Construction of the 2000 Decennial Summary Files" specification.

Every summary file contains a Geographic Header Record file, which contains the Census geocodes for each geography in the product. The Geographic Header Record is discussed in detail in the Census Technical Documentation released with each product.

5.1.4. Logging Infrastructure

As a batch processing system, DPP implements an extensive logging infrastructure – the execution details (who, what, when and where) and detailed output of every program in the system is logged. The logs are an invaluable aid when debugging code or data problems.

5.1.5. Security Infrastructure

DPP leverages AIX user and group security settings extensively to implement file and directory-based security. Standard Census "JamesBondID" user accounts do not have write access to any DPP work area. Only a set of generic, su-only accounts (dppdev, dpppa, dppuat, dpptest, dppprod, dppsprod) have write access.

All DPP software must be run as one of these named su-only accounts. This provides traceability (since the UNIX ID of every su-only user is tracked) and security (because all the files in a DPP work area belong to a single su-only user).

DPP uses work groups to allow assignment of rights and permissions to sets of users rather than to individuals. Users can be added or removed from groups without having to change the properties of objects. "Reviewers" are a special category of user. The ability of "reviewers" is limited to read-access to products that have been prepared through the DPP system.

5.1.6. Tabulation

The core processing activity in DPP is tabulation – specifically, the cross-tabulation of one or more recoded demographic variables, for a specific universe, for a specific measure (also called a summation option), for an optional characteristic iteration, for a large set of geographic areas (usually, all the geographies in a state). Each cross-tabulation corresponds to a single matrix table specification defined by POP/HHES.

DPP performs tabulation with **SuperSERVER** on AIX. During a production run, the template TXDs (created by Subject Matter Experts) are merged with a geographic recode and a characteristic iteration recode (for iterated products) to create a set of merged TXDs. The SuperSERVER engine processes batches of merged TXDs. For each merged TXD, SuperSERVER creates a CSV output file, with one line per geography. The CSV has N+1 columns, separated by commas – the GEO_ID, and the N cells of the matrix table. The CSV files may be subject to further post-processing.

Logically, the DPP system can distribute the processing of tabulation requests across multiple CPUs and physical machines, and then bring the results together to make a single set of Summary Files. This ability to segment tabulation work across multiple machines was essential to complete the enormous set of Decennial Census 2000 tabulations on time.

5.1.7. Post Processing

Depending on the product specification, a number of disclosure avoidance, thresholding, or rounding calculations are applied to the data after core tabulation by SuperSERVER on AIX. These post-tabulation activities are generally done with SAS. Each intermediate transformation of the tabulation output is saved on disk, and preserved as an audit trail that can be used later to examine each data transformation step.

5.1.8. Quality Assurance

All summary files produced by DPP are reviewed for correctness by the DPP staff and for correctness and disclosure issues by staff from POP and HHES. The DPP staff executes a series of internal quality checks to verify the correctness of the summary files, using internal consistency checks (called Internal Matching), consistency checks with previous products (called Prior Product Matching), and consistency checks with independent tabulations from DSCMO (called Analyzer Matching). Selected parent-child summary level pairs (for example, states and counties) in the summary files are checked to make sure the sum of the total population of the children matches the total population of the parent. This Verify Rollup process is an independent check that the block decompositions in the geographic recodes are correct.

The DPP staff also verifies the Land and Water area control counts on the geography files by performing a special QC ("quality check") tabulation of the Land and Water area embedded in each block record using the geographic recode. These Land and Water control counts are independent tabulations embedded in the geography file by the Geography Division. The Land and Water Area for each geography is the sum of the Land and Water area of its consistent blocks. These control counts are a critical consistency check that DPP has processed the block decompositions correctly.

5.1.9. Handoff

After the DPP staff verifies the summary files, they're handed off to AFF via a shared file system. AFF loads the files into Internal Review and POP/HHES analysts review the data. After the summary files are approved, they're released to the public on American FactFinder, and delivered to ACSD for CD-ROM production, and transferred to SSD for deployment on the Census public FTP server.

5.1.10. Design Influences

In 1998, DADS evaluated Oracle, SAS, and SuperSTAR as commercial off-the-shelf software to perform the Decennial 2000 tabulations. Based on its superior performance, native cross-tabulation support, and powerful Windows GUI client for composing tables, **SuperSTAR**, from the Australian company **Space Time Research** (**STR**), was chosen as the tabulation engine for DADS. SuperSTAR I was used for the 1998 Dress Rehearsal. During this period, IBM contracted with Space Time Research to port their product to AIX in anticipation of the production workload for 2000. During the 2000 tabulations, DADS also upgraded from the Windows-based SuperSTAR I to the AIX-based SuperSTAR II. The SuperSTAR product suite consists of three main components:

- **SuperCHANNEL** - a database building utility that creates SXV4 databases,
- **SuperCROSS** – a Windows client to create TXDs and perform ad-hoc tabulations against remote SXV4 databases,
- **SuperSERVER** – a UNIX process that performs tabulation with SXV4 and TXD inputs.

All DPP inputs and outputs are either ASCII text files or SAS files. Text formats have a number of advantages – they're easy to read with SAS, the format is transportable across operating systems, and they can be read without special software for many years into the future.

Internally, the DPP system converts the hierarchical microdata text files into proprietary SuperCROSS **SVX4** database format. The core tabulation software – SuperCROSS on Windows (used for TXD creation and ad-hoc tabulation) and SuperSERVER on AIX (used for batch production tabulation) – operates against microdata stored in SXV4 databases. SXV4 databases are highly optimized for large-scale tabulation and store microdata records column-wise, instead of row-wise, as in a traditional relational database. This column-based approach accounts for SuperSERVER's remarkable tabulation speed.

SuperSERVER performs tabulation based on detailed instructions conveyed in the TXD file. Creating a TXD in SuperCROSS consists of several activities centered on variable recoding:

- Define a set of demographic variable recodes to satisfy the matrix table definition,
- Define a cross-tabulation of these recoded demographic variables,
- Create recodes to define the table's universe (the universe defines *who* we're counting, e.g., persons over 18),
- Create recodes to define the table's **summation option** (the summation option defines *what* we're counting, e.g., income, population, etc.).

In addition, during tabulation, one or two external recodes are merged into the template TXD to create the production TXD:

- Characteristics iteration recode (only for iterated products),
- Geographic recode (for all products).

The geographic recode is usually a very large file that lists the constituent blocks for each geographic area in the product. This is necessary because the SXV4 database ultimately contains only person and housing records at the block level, so all tabulation instructions need to be expressed in terms of which blocks to include in each tabulation. Geographic recode files are also called block equivalency files, and they map every GEO_ID in a product to its constituent blocks. SuperSERVER executes each TXD repeatedly, once for each block equivalency mapping defined in the geographic recode file. Geographic recodes play a central role in DPP and are produced by the ProcessGeo script (described below).

The fact that Census geography is hierarchical with well-defined parent-child relationships is not used in DPP (other than the VerifyRollup quality assurance program). Every geographic area in DPP is decomposed into its constituent blocks in the geographic recode. In this sense, there's no difference between states, counties or block groups in DPP – they're all aggregations of block level data. This uniform approach simplifies the geographic processing.

The SuperSERVER tabulation engine is a 32-bit application. Because SuperSERVER builds the tabulation cube entirely in memory, it is possible that very complex tabulations may exceed the 2 GB memory limit of a 32-bit process. Some factors that can cause a single tabulation request to exceed the 2 GB user memory limit are the number of records in the SXV4 database, the complexity of the TXD, or the size of the geographic recode.

A number of innovative “divide-and-conquer” approaches were developed in DPP to side step the 32-bit nature of the software. For example, for the SF1 product – which consisted of 52 individual state summary files and one national summary file – the DPP system created 52 state-based SXV4 databases, and then used SAS to aggregate the national results. This was necessary because building and tabulating against a single national level database (with 8 million blocks and every person and household microdata record) would exceed the 32-bit limit. Another practical consideration is the data for HDF and SEDF arrives on a flow basis state-by-state, so it makes sense to tabulate at the state-level.

Another memory-saving technique is to break the geographic recode file into segments, tabulate each segment separately, and then combine the results. A related workaround is to break extremely complex tabulations into smaller TXDs. This technique was used for the median table PCT42 – MEDIAN NONFAMILY HOUSEHOLD INCOME IN 1999 (DOLLARS) BY SEX OF HOUSEHOLDER BY LIVING ALONE BY AGE OF HOUSEHOLDER [15] – in Summary File 3. The table was broken into three smaller TXDs that were tabulated separately and the results combined in post-processing.

5.1.11. Median Processing

Medians are relatively difficult to compute on large record sets. For that reason or perhaps for other statistical reasons, the Census Bureau implements median calculations based on applying a mathematical function on a frequency distribution defined in the product specification.

Census medians are handled in one of two ways:

- Non-income related medians are computed using a linear interpolation algorithm on the frequency distribution
- Income-related medians are computed using a logarithmic interpolation algorithm on the frequency distribution called **Pareto**. The **Pareto** algorithm is unique to Census.

IBM contracted with Space Time Research to incorporate the Census Pareto algorithm into the SuperCROSS and SuperSERVER products as a supported derived measure.

For state-based products, a single TXD can compute both the frequency distribution and the Census Pareto median in one step.

Since DPP does not build a single US SXV4 database, a different approach is required to compute US medians. The following approach is used:

- DPP aggregates the state-based frequency distribution CSV files using SAS (since medians are non-additive, simply aggregating the state-based medians is incorrect).
- The aggregated frequency distribution CVS files are input to SuperCHANNEL to build a US frequency distribution SXV4 database.
- Special TXDs are created to tabulate US-level medians against the frequency distribution SXV4 database.

5.1.12. DPP Logical Application Architecture

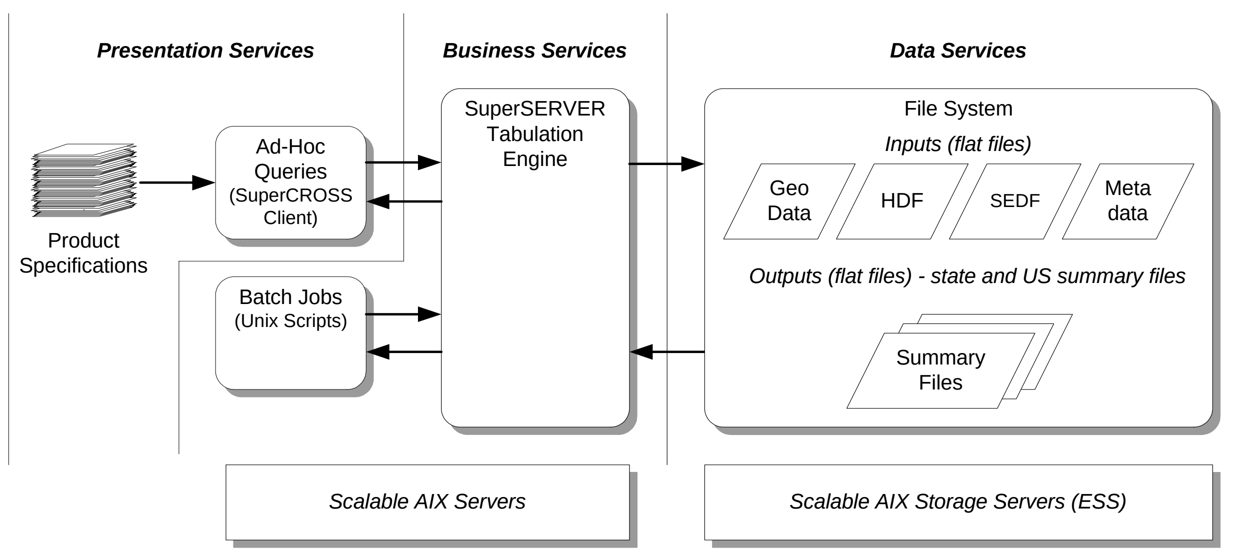

Figure 17: DPP Logical Architecture Overview

The DPP system is a general-purpose data product production system. It produces data products based on product specifications defined by POP/HHES, geography data from Geography Division, and microdata files generated from the 2000 Decennial Census. Each product has specific business requirements that generate reusable architectural capabilities in the DPP system. A key design decision was to abstract and parameterize these architectural capabilities into text-based driver files.

Subsequent products with the same business requirements can then leverage these capabilities by setting the proper driver files attributes. This parameterization of the system is called **operational programming** and the text-based inputs are referred to as **operational material**. The set of DPP operational material includes the following –

- **Driver Files** – DPP defined text files used by DPP scripts to parameterize their behavior on a product-by-product basis
- **TXDs** – textual table definitions (text-based SQL-like queries against an SXV4 database that define a cross-tabulation)
- **Match Specifications** (for prior-product, internal, and analyzer matching)
- **Analyzers** (independent tabulations) – supplied by DSCMO
- **Characteristic Iteration Recodes** – defined in the product specification and created by DPP subject matter experts as TXD fragments

The Census Bureau owns all the operational inputs to the system, and is responsible for their completeness and accuracy.

The runtime instructions for the DPP system are defined in the **DPP Production Cookbook** – a "how-to" manual that defines the runtime sequence for every DPP product – from P.L. 94-171 through SF4, AIAN, and the School District special tabs.

The interface to the DPP system is the UNIX command line. The DPP Operational Staff is responsible for creating UNIX job control scripts that implement the Cookbook instructions. Users interact with the DPP system via **DPP scripts** – UNIX Korn shell scripts, SAS programs, and various SuperSTAR scripts – that collectively define the application architecture for DPP, and provide the infrastructure to create data products – building SXV4 databases, processing geographic inputs, error handling, parallel processing, logging, security, tabulation, post-tabulation, summary file creation, quality assurance, and handoff capabilities to AFF, POP, and ACSD.

In preparation for production tabulation, the DPP Operational Staff define, create, and test the textual table definitions (TXDs), as per the production specification, using the SuperCROSS client. To support the TXD sourcing effort, the DPP Development Team creates and publishes an initial “sourcing SXV4 database” from the HDF or SEDF microdata. Depending on the product specification, extra fields may be added to the HDF and SEDF microdata records (such as “utility” recodes, or extra fields from the block record in the DPP Geography file). The Operational Staff perform ad-hoc tabulations to iteratively define and unit test each TXD definition. When complete, the TXDs are put under source control, along with the DPP driver files and CIs, and become part of the DPP production system.

The DPP development team creates SuperCHANNEL build scripts to convert the microdata into SXV4 databases; the DPP Production team then builds the SXV4 databases as part of the production process. Only SXV4 databases created by the Census Bureau are used for production tabulation – the development versions are used only to “bootstrap” the TXD sourcing work.

The two main DPP scripts are **ProcessGeo** and **Tab**. Each is broken-down into well-defined steps called stages.

The main function of **ProcessGeo** is to create the geographic recode for tabulation from the DPP Geography File. Its other major function is to support a number of performance enhancements that break the geographic recode into smaller, more manageable parts. The **ProcessGeo** stages perform the following functions:

- **Stage 10: Build Geographic Recode** – Process the DPP Geography File to create the geographic recode.
- **Stage 40: Reduce Recode** – Remove geographies from the geographic recode that contain no blocks (these extraneous zero-block entities consume space in the in-memory tabulation cube).
- **Stage 45: Dehydrate Recode** – Identify and remove duplicate geographies (i.e., geographies with different summary levels but the same block decomposition). A surprising percent of all the Decennial geographies are duplicates (nearly 60%). A representative duplicate geography is tabulated once, and then “re-hydrated” (reassembled) in post-processing in **Tab stage 3020**. **Deydration** is not appropriate for the School District special tabs, where the tabulation rules vary by summary level or **geosets** (so even physically identical **GEO_IDs** with different summary levels may need to be tabulated with different rules).
- **Stage 50: Split Recode** – Split a geographic recode into N smaller parts. Used solely to circumvent the 2 GB memory limit of SuperSERVER. Used infrequently (e.g., the geographic recodes for Florida, California and New York were split during SF3 processing).
- **Stage 60: Create Iteration Recodes** – Create iteration-specific geographic recodes. Remove geographies that don’t contribute to the current characteristic iteration. A separate process called SIPHC is run to determine, for each characteristic iteration, which geographies can be dropped. The SIPHC process is described in detail in the DPP Production Cookbook. SIPHC is the cornerstone of the thresholding subsystem in the DPP system.
- **Stage 70: Split Recode into GeoSets** – Segment the geographic recode by defined summary level groupings; required by the School District products, where tabulation rules can vary by *summary level*. Technically, this step is not an optimization, but a mandatory step required by the business requirements of the School District special tabs.

These optimizations work together, so it’s possible to reduce, dehydrate, split, and create iteration recodes. The Tab script in stage 3000 makes sure the geographic recode segments are tabulated and reassembled properly to create the final summary file.

The other main DPP script is called **Tab**. It also uses stages. The **Tab** stages perform the following functions:

- **Tab stage 1000** verifies the operational inputs and prepares them for SXV4 database creation

- **Tab stage 1025** assembles data for a US database build (only the AIAN data product uses a National database)
- **Tab stage 2000** builds the SXV4 detail databases
- **Tab stage 2050** publishes the SXV4 databases, with appropriate security, for remote, ad-hoc tabulation with SuperCROSS; SuperSERVER uses the same databases for batch tabulation.
- **Tab stage 3000** analyzes the list of tables to tabulate, merges the template TXDs with the geographic recode and characteristic iteration recode (if any) into production TXDs, groups them into batches, and submits the batches to the SuperSERVER tabulation engine. The production TXDs are transient and are deleted immediately after tabulation. Tab stage 3000 is also responsible for reassembling the individual CSV outputs from each geographic recode segment into a single CSV file. This step is necessary only if the geographic recode was segmented in ProcessGeo.
- For products with a national summary file and state summary files (e.g., S1, S2, SF3, and SF4), the national and state geographies are tabulated as part of the state tabulation. In other words, there is no separate "national" level tabulation – all the national geographies are computed during state tabulation (with the exception of US-level medians, which are non-additive and require special processing, as described earlier). The state and national geographies are moved into separate CSV files in **Tab** stage 3050 (described below). Note that a given geography can appear in both the state and national summary files (e.g., Alabama and all its counties are in both the state and US summary files).
- **Tab stage 3010** performs housing keeping on the CSV output file name and appends the stage names (3010) to each CSV file – e.g., *P10_U_NY.csv* becomes *P10_U_NY.csv.3010*. All the following stages append the stage name to their output as an audit trail. A key DPP design principle is that no program should overwrite its input. For traceability and restartability, every program has distinct inputs and outputs. Restartability refers to the requirement that the DPP operator should be able to restart a failed Tab stage from the point of failure.
- **Tab stage 3020** rehydrates the duplicate geographies back into the CSV output. This step occurs only if the user ran the ProcessGeo script stage 45 that dehydrates (removes duplicates from) the geographic recode.
- **Tab stage 3025** merges or overlays results from "associated" products. Associated products are used to tabulate geographic component 49 – Rural Farm.
- For products with a national and state summary files, Tab stage 3050 splits stage splits the state and national contributions into separate CSV files.
- **Tab stage 3200** aggregates the state CSV outputs for the US product (this stage should be run only after all 52 state products have been created)
- **Tab stages 3300, 3325, 3350, 3400,** and **3500** are used to tabulate US medians.
- **Tab stage 3600** splits combined tables into individual product tables. In a few cases, individual matrix tables are combined into a single combined TXD for tabulation purposes. This stage breaks the combined table CSV output into its constituent parts. This technique is most commonly used for race-iterated tables.
- **Tab stage 3700** applies unconditional rounding. Driver files control which tables and cells are subject to rounding.
- **Tab stage 4000** creates the summary file, including the geographic header record. For iterated products, the geographic header record is created only for iteration 001.
- **Tab stage 4020** and **4040** apply additional thresholding and special tab rounding rules to the Summary Files. Used only for the School District special tabs.

- **Tab stage 4200** runs the Verify Rollup program on the Summary File. The summary level pairs to check are defined in driver files.
- **Tab stage 5100** runs the Analyzer match program on the Summary File. Not all data products are supplied with analyzers.
- **Tab stage 5200** runs the Internal match program on the Summary Files. The inter-table/cell expressions to check are defined in driver files.
- **Tab stage 5300** runs the Prior-Product match program on the Summary File. The inter-product table/cell expressions to check are defined in driver files.

The Tab script is designed to run across multiple CPUs and machines. Each runtime instance of Tab is called a wave, and the DPP Production Cookbook contains extensive instructions on how to "wave" a data product for maximum runtime efficiency.

5.1.13. DPP Logical Data Structure

The conceptual architecture of the HDF and SEDF databases is similar. Both designs relate Block Records to Housing Records and the Housing Records to Person records in a parent-child relationship.

The metadata tables for each (HDF and SEDF) are different because HDF and SEDF have different attributes in the microdata. The microdata data and metadata are inputs to the SuperCHANNEL database build utility that stores the microdata in a SXV4 database optimized for tabulation.

5.1.13.1. Geographic Data

The product's sponsor (usually POP/HHES) defines the geographic coverage of a data product.

For each data product, Geography Division (GEO) delivers 52-state geographic tabular data files and one US geographic tabular data file. The file format is defined by DPP in the DPP Geography File (DGF) specification. The DPP team, in conjunction with GEO, defines the rules to determine the constituent blocks for each geographic area in the product. These rules are used by the ProcessGeo program to build the geographic recode file. The DPP system encodes the business rules in the DPP driver file called GeoIDInfo.txt.

5.1.13.2. Metadata

DPP receives documentation of the valid values for each field in the HDF and the SEDF. This documentation (a spreadsheet) is broken into individual ASCII files by attribute; these files identify the valid code values and the code descriptions of the microdata attributes. The files are loaded into tables and are used to validate the microdata. They form the valid value or "classification" tables in the SXV4 database. When loading microdata records into the SXV4 database, any record with attribute values not in the corresponding classification table is rejected and causes the database build to fail.

5.2. DPP Physical Architecture

The DPP system tabulates census microdata provided by BOC data providers based on business rules defined by the product specification. These rules are designed to both provide useful summary data to the public, as well as protect the confidentiality of survey respondents, as required by Title 13. The Title 13 nature of the microdata played a leading role in the physical architecture and security setup of the DPP system.

The DPP AIX server hardware and network are located in the Bowie Computer Center Mod 2. The DPP LAN server hardware and operator workstations are located in Suitland. More detail on the physical architecture can be found in the DPP Technical Design document.

5.2.1. Design Influences

Some of the significant design influences include:

- The DPP system houses Title 13 microdata. The need for fine-grained security motivated the use of UNIX instead of Windows as the operating system.
- All work on the DPP system is ultimately focused on creating accurate summary files as quickly as possible. Due to the sheer size of the Decennial products, production processing for a single product can run continuously for days, weeks, or in some cases, months on end. Therefore, virtually all products require multiple executions of the DPP system. These massive products motivated steady improvements and optimizations in the DPP system to leverage multiples CPUs and machines to perform as much parallel processing as possible. The high-end computing nature of tabulation and data processing drove the primary technology choices:
    - Three IBM pSeries p680 with 24 processors and 96 GB of memory (two of these p680s are now part of the AFF external environment).
    - IBM Enterprise Storage System (ESS) with over 10 TB of disk space.
- The DPP system generates an enormous number of both temporary and permanent files. Some files are subject to intense I/O processing. The configuration of the AIX JFS (journal file system) was designed on a product-by-product basis based on the expected number of files in the system. Files expected to experience heavy I/O were placed on separate stripped disks.
- The DPP system makes extensive use of NFS to allow the production machines to have the same logical view of the file system. The NFS file systems are run over a Gigabit Ethernet router, and the NFS parameters are tuned for the DPP usage pattern (lots of read-access to small files).
- The results of tabulation are subject to review by BOC data provider subject matter experts (SME) for correctness and Title 13 disclosure issues. After review, SMEs can determine that the BOC data provider and/or DPP need to modify the data and/or the tabulation method. This delivery-tabulation-review cycle can be repeated multiple times for the same census data product.

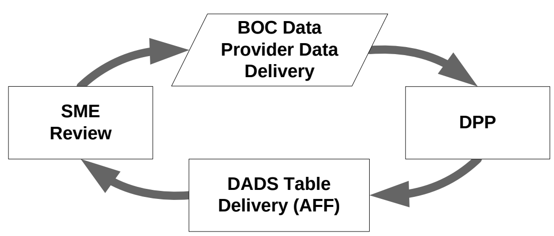

Figure 18: DPP and the Iterative Processing Cycle

- The major Decennial tabulations are used by an extremely wide-range of constituents. For example, PL 94-171 provides redistricting data needed by the 50 states for their use in redrawing districts of the United States Congress and state legislatures. A number of Federal entitlement programs are indexed to data from the long and short form data. As a result, the Census Bureau publicly announces the expected release date of the major Decennial products. Any delay in the tabulation of these products reflects badly on the US Census Bureau.

DPP, in order to meet the challenges and requirements listed above, must:

- Be configurable and extensible using a layered architecture
- Be scalable to perform multiple tabulations at once
- Define an architecture to support the distribution of tabulations across multiple physical machines (horizontal scaling)
- Handle very large data volumes from multiple sources
- Use existing DADS system security systems (no external users)

In order to meet temporary peaks in processing, DPP must have support an “On-Demand” architecture that includes the ability to reposition hardware and other resources to and from DPP during peak processing periods. For example, two of the high performance IBM P680 servers supporting AFF were loaned to support DPP production for Census 2000.

5.2.2. DPP Physical Overview

The DPP hardware consists of three servers (a development UNIX server, a production UNIX server, and a Windows-based file server), a storage server, and multiple Windows-based developer workstations as shown in **Figure 19.** DPP Physical Overview.

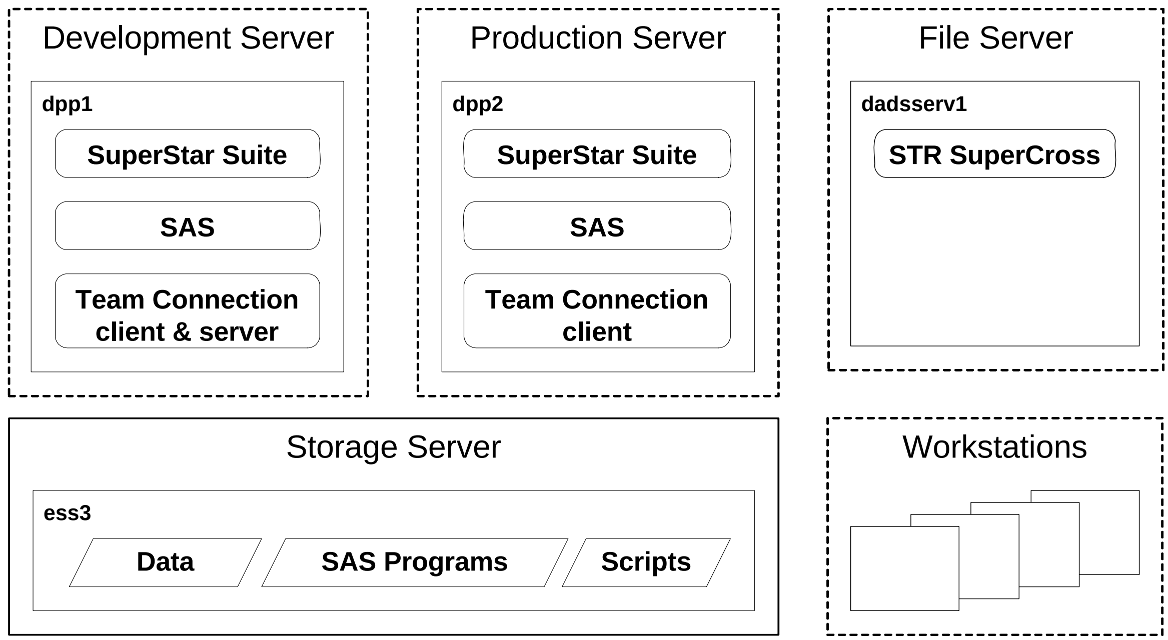

Figure 19. DPP Physical Overview

The subject matter experts use the **STR SuperCROSS** software, stored on **dadsserv1**, the **File Server**, to create tabulation **TXDs**. A **TXD** is a text-based file format, defined by **Space Time Research**, which defines a "textual table definition"– a SQL-like statement to compose a specific tabulation. The tabulation **TXDs**, along with the **Scripts** (both **SAS Programs** and **Korn shell scripts**) are tested using the **STR SuperSERVER** applications on **dpp1**, the **Development Server**. When the tabulation **TXDs** and **Scripts** are approved for production, the tabulations are performed using the **STR SuperSTAR** applications on **dpp2**, the **Production Server**.

The **TXDs** and **Scripts** themselves are stored on **ess3**, the **Storage Server**, using the **Team Connection** change management software. The **Data** used by, and created by, the tabulation **TXDs** and **Scripts**, are also stored on **ess3**. The **Data** are imported and exported from the DPP system using **SAS** as directed by **SAS Programs** stored on **ess3**.

In order to handle the data volumes and processing requirements, the **Development Server** and **Production Server** are large, multiprocessor AIX systems with large memory configurations. If needed, the architecture allows additional **Production Servers** to be added to the DPP system during peak processing periods in order to handle complex tabulations and to meet schedule milestones.

To handle the large volumes of data, the **Storage Server** is connected to the other servers using a high-speed interface. To ensure system availability, the **Storage Server** uses RAID storage. To ensure system recoverability, the DPP system is connected to an automated tape backup system.

Multiple networks are used by the DPP system to meet performance requirements and to isolate the data for security reasons.

5.2.3. DPP Software Deployment

DPP is implemented using a combination of:

- Custom AIX Korn Shell scripts,
- SAS software, and
- Space Time Research SuperSTAR tabulation production software suite.

All operations are performed on IBM servers running the AIX operating system.

DPP software is divided into three categories:

- **Custom** – Applications and databases written to support a system function or enhancement
- **COTS** – Vendor-produced software, also known as Commercial Off-The-Shelf (COTS) software
- **Server** – Applications supporting the operation and administration of the servers and network

The following table is a high-level matrix of the DPP software and databases grouped by the above categories:

	Software	Description
Custom	Korn Shell scripts	Batch job programs, control programs, and system/database administration
	Perl scripts	Batch job programs
	SAS programs	General purpose data processing
	TeamConnection DB (Census)	Holds the tabulation control files, including driver production files and detail table definitions files (TXDs), within the TeamConnection configuration management system
	HDF SXV4 DB Files (HDF*U.sxv4)	HDF SXV4 databases built with SuperCHANNEL build utility from HDF microdata input
	SEDF SXV4 DB Files (SEDF*U.sxv4)	SEDF SXV4 databases built with SuperCHANNEL build utility from SEDF microdata input
COTS	STR SuperSTAR Suite Build 53	Provides DPP production and development tools, including: **snbu** – Command line SXV4 database builder tool **scstools** – Command line tool to create database catalogues **scs** – Command line tool to start server associated with a catalogue file **SuperCROSS** – Windows application to compose tables and perform manual tabulations **ss2ps** – Command line batch tabulation engine **python** & **jpython** – Suite support tools
	SAS v8	Data processing tool
	ViewNow v1	AIX server access tools including X-Windows, telnet and FTP (Windows)
	IBM Java JDK v1.3	Operating environment needed to support SuperSTAR suite
	IBM TeamConnection v3	DPP source code control and configuration management system (AIX and Windows). There is one TeamConnection server running on dpp1. The TeamConnection client software is installed on all the DPP UNIX and Windows machines. The clients all access the same TeamConnection server.
	gzip (open source)	Data compression tool used to compress summary files for handoff to AFF
	PKZip v2.5	Data compression tool used to compress summary files for handoff to ACSD
System	AIX v4.3.3	Server operating system
	TSM Server v5.1	Tape back-up system software for servers and clients
	TSM Client v5.1	

Table 5: DPP Software

The following figure shows the distribution of the above software on the DPP hardware.

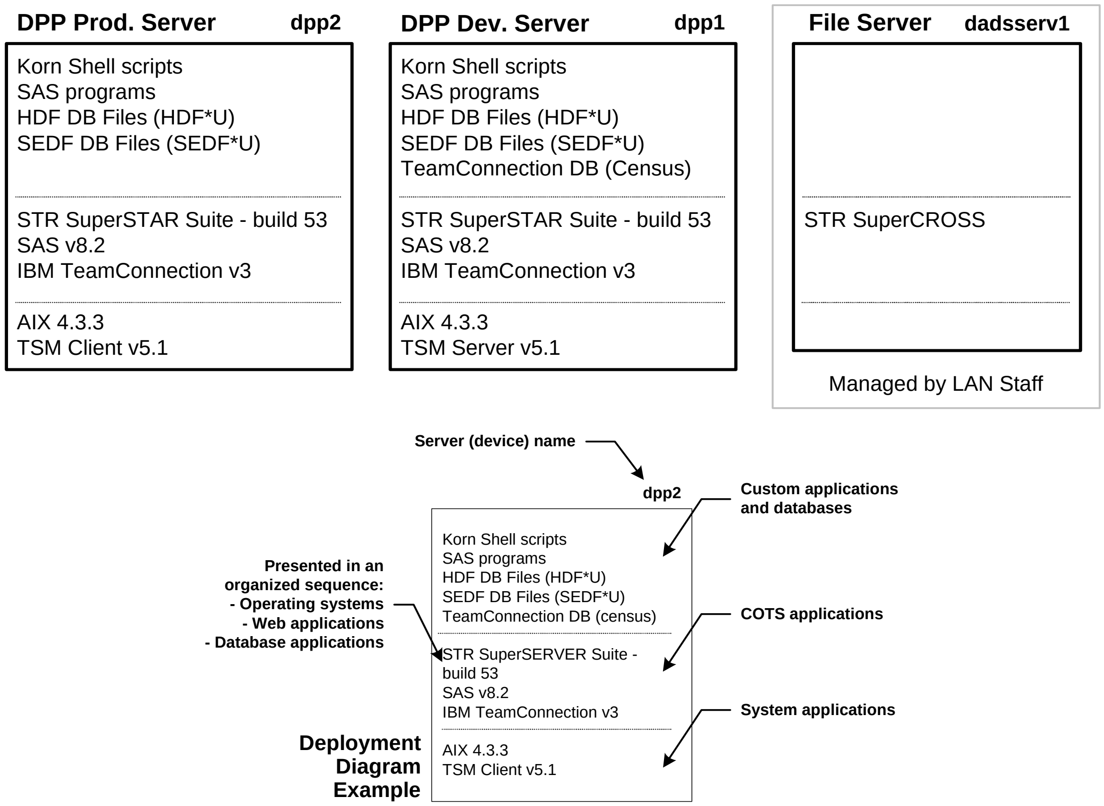

Figure 20: DPP Software Deployment

The above figure does not include the various Windows workstations used by developers and testers to develop, execute and test tabulations. These users use a network installation of the STR SuperCROSS from the dadsserv1 file server and local installations of ViewNow and IBM TeamConnection client.

5.2.4. DPP Operations Scenarios

The DPP Production Cookbook defines all the operational runtime sequence for the DPP system. The runtime sequence is developed and documented by the DPP Development Team, verified by the DPP Product Assurance Team, and independently verified by the DPP Operations Team in user acceptance testing (uat). The DPP Operations team is responsible for the production runtime operation of the DPP system, and all Quality Assurance and Handoff procedures.

Logically, the DPP system is run once and only once for each data product. Once the summary files are created and approved, the DPP system is never run again for that specific data product. In this sense, each DPP data product is a one-time undertaking. The specific runtime sequence varies from data product to data product, but follows the general pattern described earlier:

- Preparation of operational material – TXDs, driver files, characteristic iteration recodes, match files and specifications for internal, prior-product, and analyzer matches
    - Note that this activity starts in development with “starter” versions created by the DPP development team. The operational files are tested and vetted as they move through the development and product assurance cycle. The DPP system contains many self-checking mechanisms that can detect errors. By the time production starts, the operational files have undergone intensive testing.

- data and metadata preparation
    - Executing the Get script to retrieve and format the inputs files from GEO and DSCMO into the DPP runtime environment.
- create geographic recodes
    - Execute the ProcessGeo script
- build SXV4 databases
    - Usually, the Product Assurance Team has already built the SXV4 databases, and any data issues have been discovered and fixed by the time production starts. For control and traceability, databases are always re-built in the Production environment.
- tabulation
    - Execute Tab stage 3000
- post tabulation
- summary files creation
    - Execute Tab stage 4000
- quality assurance
    - Execute Tab stages 5200, 5300, 5400
- handoff
    - Execute the Handoff script

5.2.5. DPP Hardware Architecture

Please reference the Attachment document for more details on the DPP Hardware Architecture.

5.2.6. DPP Network Architecture

Please reference the Attachment document for more details on the DPP Network Architecture.

6. OTHER DADS SYSTEMS

In addition to AFF, AQ and DPP, there are several other DADS systems used in the implementation, presentation and monitoring of DADS. These systems include:

- Virtual Table Generator
- PHC Reporting System
- Beyond 20/20
- Web Trends Analysis Suite

This section briefly describes these other DADS systems.

6.1. Virtual Table Generator (VTG)

The Virtual Table Generator (VTG) is a Java-based GUI tool that produces three main groups of file products: HTML review files, metadata load-able files and SQL view files. Its primary role is to assist table designers in the design and specification of virtual tables. The VTG may be accessed from any BOC workstation mapped to the I:\ drive.

6.2. PHC Reporting System

The Population Housing Characteristics (PHC) Reporting System is an extension of AFF providing publication quality report generation to the BOC Population Division. POP further integrates the reports for assembly into publication-ready reports. The Population Division defines the reporting characteristics and is responsible for dissemination of the PHC reports to the public.

The PHC system leverages existing AFF functionality to build complex tabular presentations, and extends its by outputting XML instead HTML files. This tool uses an application maintained by Fenestra Technologies Corporation, based in Germantown, Maryland. Custom code in AFF merges formatting information and tabular data into an XML file. The Fenestra tool takes the XML file, along with additional control information, and generates a PDF file. The PDFs are delivered to the Population Division for assembly into publication-quality books.

6.2.1. Architectural Highlights

The PHC Reporting System uses extensive XML coding to define the contents of PHC reports. Architectural design highlights include:

- A customization of the AFF application and a Virtual Table Generator (VTG) available from any BOC workstation mapped to the I:drive.
- XML property and data file management using the commercial application, Fenestra.

The VTG generates metadata including PHC TXT files and stores the data in the AFF MDR and Data Warehouse. PHC reports are assembled on the DADS Property Generator Workstation in Suitland, using the Fenestra application.

Final report assembly, also using the Fenestra application, combines AFF metadata, XML data and a selected report format. This assembly is managed on the DADS AutoFormatter Workstation in Suitland. PHC reports are presented as PDF files, and are posted on an internal AFF web site for review.

The Fenestra Technologies Corporation, Germantown, Maryland, supplies the Fenestra AutoFormatter application.

6.2.2. Hardware and Network Architecture

The DADS Property Generator Workstation and the DADS AutoFormatter Workstation, located in Suitland, support the PHC Reporting System. Both workstations are DELL PW S530 P3 using MS Windows 2000. These machines reside on the BOC Intranet.

6.3. Beyond 20/20

The American Community Survey (ACS) Beyond 20/20 Pilot project was launched in July, 2003 to assess the capabilities of the commercial software application, Beyond 20/20 WDS. The project is organized to meet a combined ACS – DADSO objective to provide supplemental Census data dissemination targeted to the special needs of reporters and members of the media regularly monitoring Census Bureau activities and data. The project's functional and technical assessment was delegated to DADS.

Beyond 20/20 Web Data Server is a customizable software application. It is the product of Beyond 20/20 Inc. with US offices in Alexandria, Virginia. The pilot project has included the installation of application upgrades and service packs. Version 7.1 is currently deployed.

6.3.1. Architectural Highlights

Beyond 20/20 WDS is a data reporting and dissemination tool with OLAP dimensioning characteristics. The application allows the user to quickly analyze information that has been summarized into multidimensional views and hierarchies, such as Census data tabulations. Beyond 20/20 WDS is MS Windows-based technology with the following attributes:

- Builds tables (or extracts) using the application's toolkit and wizards
- Provides multiple dissemination capabilities including a custom browser, server-supported Internet and Intranet publishing over a LAN or from an FTP site, tables duplication onto CD-ROM.

6.3.2. Hardware and Network Architecture

Beyond 20/20 is currently supported by the following hardware:

- **Web/ Application Server – dadso01acs:** DELL 2450, two-way, 2.8GHz P4 processors with 2GB memory running MS Windows 2000.
- **DB Server – dadso02acs:** DELL 6300, four-way, 450MHz P4 processors with 2GB memory running MS Windows 2000.

Beyond 20/20 servers are currently networked in tandem at DADS NET: 148.129.77.210 and are located in Mod 2 Cage at the Bowie Computer Center.

6.4. Web Trends Analysis Suite

Commonly referenced as, "Web Trends," this commercial software application is used by DADS as a Web traffic analysis and reporting tool. Web site activity is reported using key indicators including:

- Number of Web site visits
- Visitor point of origin
- Web page preference

The analysis suite also provides internal system performance and system use analysis including proxy traffic analysis, system monitoring and alert processes, and Web site link analysis. Web site load balancing and mirroring may also be monitored.

Web Trends Analysis Suite, Advanced Edition, v7 is MS Windows-based technology and currently deployed on two Suitland DADS workstations. Access is ID and password protected.

The application's vendor, NetIQ Incorporated is headquartered in San Jose, CA.

6.5. Other Software

In the development environments, additional software is used to build and maintain AFF, AQ and DPP; the following table is a high-level matrix of the additional software and databases used for development and testing:

Software	Description
ETL Shell scripts	AFF system and database administration programs
Rational Schema DB	ClearCase and ClearQuest databases
IBM ClearCase Client v2003	Manages versions of files and directories – configuration management tool
IBM ClearQuest Client v2003	Defect and change tracking system – change request management tool
Informatica PwrConnect v7.1	ETL tool – delivers data to AFF Data Warehouse
Oracle Enterprise v9.2	Database system development tools – multi-node parallel storage clusters
Toad v8	Database query tool (Windows)
ViewNow v1	AIX server access tools including X-Windows, telnet and FTP (Windows)
LoadRunner	Automated test tool (Windows)
ERWin	Database design and maintenance tool (Windows)
XMLSpy	Used to create and edit ESRI ArcXML files (Windows)
ArcView	Part of ESRI product suite; used to verify contents of spatial database
ArcGIS	Full ESRI desktop suite; used to verify contents of spatial database

Table 6: DADS Development and Test Tools

7. DADS OPERATIONAL OVERVIEW

As depicted in the previous sections, the DADS architecture is comprised of multiple hardware components, software components, and networks. These items are housed at the Bowie Computer Center. Workstations, terminals, networking devices, and personnel within the DADSO facility in Suitland and at other BOC and non-BOC facilities support them.

DADSO is responsible for the design, development, maintenance, and administration of the DADS applications, excluding some of the shared resources owned by other BOC entities. DADSO is also responsible for the data processing that populates DADS databases and metadata repositories.

DADS has been operational since 1999 to support the first stages of the Decennial 2000 Census.

7.1. Bowie Computer Center

As shown in **Figure 21:** DADS Deployment - High-level View, DADS is primarily located at the BOC Computer Center in Bowie, Maryland. The hardware and networking infrastructure that support DADS (i.e., AFF, AQ, DPP, and Beyond 20/20), includes equipment supplied and managed by the Bowie Computer Center such as workstations, cabling, firewalls, hubs/switches and related network devices.

7.1.1. Module Configuration

The BOC Computer Center is organized into seven modules, referenced as “Mods”, which may be configured differently to support specific requirements. The BOC Computer Center distinguishes two environments within each Mod: “Internal” or “External”. “External” system elements are gated and locked in a “Cage,” to ensure security. Each cage is a DMZ (demilitarized zone), which controls access between the Internet and a secure LAN. All publicly accessible systems are restricted to a DMZ. The “Internal” environment supports access from the BOC Intranet.

The AFF, AQ, DPP, and Beyond 20/20 systems share Mod 2. Most AQ system elements, however, are housed in Mod 6. The operational systems are supported by the system elements housed within the cages except for DPP system and the Title 13-sensitive infrastructure for AQ.

- **Mod 2 Cage** – Initial entry point for DADS external Internet users. The Mod 2 Cage houses the Edge, Busy, and Unavailable servers, which support both AFF and AQ. The hardware and networking infrastructure for AFF and Beyond 20/20 are contained in this cage as well as the AQ Web servers.
- **Mod 6 Cage** – Supports AQ. The AQ Intelligence servers and Database servers that support the external Internet users are housed in this cage.
- **Mod 6** – Supports AQ. Mod 6 is accessible to BOC Intranet users only. Web, Intelligence, and Database servers that support both the AQ operational system (for internal BOC users) and other DADS environments (e.g., AQ development environment) are contained within this mod.
- **Mod 2** – Supports DPP and AFF. The operational DPP system as well as other DADS environments (e.g., development) for DPP and AFF are contained in this mod.

7.1.2. Bowie Computer Center Resources

The facilities at the BOC Computer Center provide the physical environment necessary to housing a system as complex as DADS. Other resources also contribute to the DADS infrastructure, most notably the Bowie Firewall. The Bowie Firewall provides the main control point for access from the Internet to the DADS. The Firewall also provides control policies specific to AQ, which requires more rigid access controls to protect its underlying microdata.

Network switches (e.g., Cisco switches from the firewall), workstations (e.g., SP control workstations), routers (e.g., Intranet router) are also supplied by the BOC Computer Center to facilitate the operation and administration of DADS.

7.1.3. DADS Environments

DADSO manages four systems that are operational: AFF, AQ, DPP, and Beyond 20/20. These operational systems are in their respective "Production" (Prod.) environments. Other DADSO environments exist to support the design and development of the systems as well as on-going data deployments. These environments are referenced as Development (Dev.), Internal Review (IR), or Product Assurance (PA).

7.1.3.1. Production Environments

DADS operational systems are accessible from both the Internet and the BOC Intranet. DADS Internet traffic is first routed through the Bowie Firewall then through the DADS Edge server to manage both security and load on the systems. BOC Intranet traffic may also be routed through the Bowie Firewall. The location of the production environment is different for each DADS system.

- AFF Production – All servers and supporting hardware is located in the Mod 2 cage.
- AQ Production – AQ has two production systems: an "external" production system and an "internal" production system.

  The AQ "external" system accesses a database that has cleansed microdata to minimize exposure of Title 13 confidential information to its authorized Internet users. The AQ "external" system is primarily housed in the Mod 6 cage. Only the Web server for "external" production is housed in the Mod 2 cage.

  The AQ "internal" system provides BOC end users with access to Title 13 confidential information. It uses the same HDF, SEDF, and metadata databases as the "external" system, which are located in the Mod 6 cage. Its Web and Intelligence servers are housed in Mod 6. Traffic to the AQ "internal" system is also routed through the Bowie Firewall, even though it originates from the BOC Intranet.
- Beyond 20/20 – Located in Mod 2 cage, this system is being hosted by DADSO for ACSO. It is accessible to authorized Internet users only.
- DPP Production – The DPP Production system is located in Mod 2. It is accessible to end users via the BOC Intranet.

7.1.3.2. Internal Review/Product Assurance Environments

Internal Review/Product Assurance environments are used to support testing of the AFF or AQ systems.

- AFF Internal Review – The system elements that make up the AFF IR environment are housed in the SP1 frame in Mod 2.
- AQ Product Assurance – The AQ PA environment has the same configuration as the "internal" system, except it uses the Development database to access HDF data. That is, PA uses the same "internal" system Web and Intelligence servers with the Development database from Mod 6 and the SEDF and metadata databases located in the Mod 6 cage.

7.1.3.3. Development Environments

- AFF Development – The system elements that make up the AFF Development environment are housed in the SP3 frame in Mod 2.
- AQ Development – The system elements that make up the AQ Development environment are located in Mod 6.
- DPP Development – The DPP Development system is located in Mod 2.

7.1.3.4. Sandbox

Located in Mod 2, the sandbox has a server in the SP4 frame. The system administrators test new COTS software, patches, or hardware configurations on the Sandbox before deploying to other environments.

7.2. Other DADS Resources

As depicted in **Figure 21:** DADS Deployment - High-level View, DADS uses resources that are not located at the Bowie Computer Center. Some of these resources are managed by DADSO and others are managed by other BOC entities or external vendors.

7.2.1. DADSO Suitland

The DADSO office in Suitland contains workstations to support administration of DADS in Bowie.

The hardware and network infrastructure for the PHC Report System is located in the DADSO office. These workstations serve as the development and production environment for the PHC Report System, which creates PDF image files of Decennial data.

7.2.2. Shared BOC Resources

The DADSO team uses the following resources to support the development, testing, and operation of AFF, AQ, and DPP:

- BOC LAN (I:\drive) – used by the DADSO team to store work products and other files, including the Virtual Table Generator (VTG) (that supports AFF).
- Census FTP Servers – used by the data providers to deliver data to DADSO (both microdata and aggregated data) as well as by DADSO to provide files to external Internet users.
- BOC Beta Testing Site – used to support load and performance testing and other quality assurance procedures.
- SuperCROSS File Server – used to house the SuperCROSS GUI that is allows authorized users access to the DPP servers in Bowie.

7.2.3. External Vendor Resources in support of AFF Performance Monitoring

Web performance monitoring has been procured from Mercury Interactive Corporation. DADS is monitored using Mercury Topaz from five sites across the continental United States:

- UUNet – Austin, Texas
- AT&T – Chicago, Illinois
- Exodus – Los Angeles, California
- UUNet – Seattle, Washington
- UUNet – Washington, D.C.

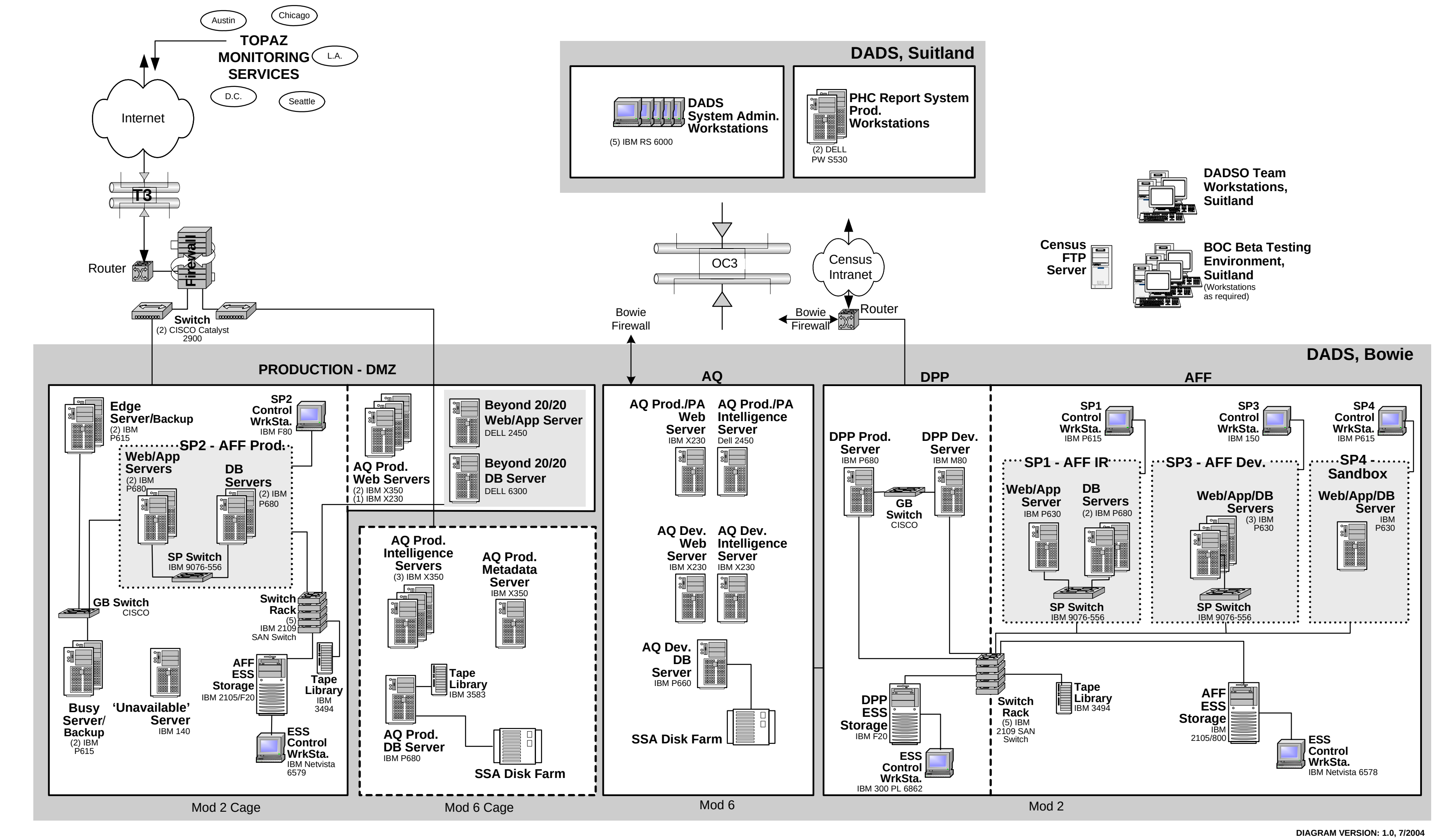

Figure 21: DADS Deployment - High-level View

7.3. DADS Data Processing

DADS data transformation processes support on-going data deployments or Decennial Census tabulations. DPP, the Virtual Table Generator (VTG) application, and Extract-Transform-Load (ETL) scripts are used for these processes. As shown in **Figure 22:** DADS Data Processing - High-level View, these processes rely on inputs from several sources, including DADSO. These processes also produce several different outputs including ASCII files, Oracle data, or IBM DB2 data.

7.3.1. Data Sources

Data is delivered to DADS in a wide variety of data formats, such as ASCII files (e.g., for product metadata), SAS files (e.g., for ACS base tables), or binary shapefiles (e.g., the spatial file deliveries from GEO). Data are provided as microdata or already aggregated data, depending on the source program. Metadata are delivered by the data owners, created by DADSO, or manufactured using VTG. The following data are delivered:

- Hundred-percent Data File (HDF) – Microdata from the Decennial short form, which was sent to each household in 2000.
- Sample Edited Detail File (SEDF) – Microdata from the Decennial long form, which was sent to one-in-six households in 2000.
- Data Files (DEC, PEP, ACS, and ECON Files) – Aggregated data from the associated program. This data are delivered in either SAS or ASCII format. The frequency of delivery depends on the program.
- Non-Spatial Geographic Data Files – Non-spatial geographic (NSG) data for each program support data processing of both micro and aggregated data. The NSG data files contain records that describe the geographic content of specific Census surveys and general orienting feature such as roads and railways.
- Spatial Geographic Data Files – ESRI shapefiles from Geography Division, as specified by DADSO. The spatial files are loaded in ArcSDE and support the three main AFF mapping features – (1) Thematic Maps (data visualization); (2) Reference Maps; and (3) Map-based selection of one or more geographic areas in support of a data query.
- Address Files – Address data that supports AFF search function is obtained from an external vendor, Sagent.
- Data Warehouse Maintained Metadata (DWMM) – Metadata produced by DADSO that support AFF application processing including navigation within the site.
- Product Metadata – Metadata from the associated program that support the rendering of a program's data or derived products through AFF. Depending on the program, this metadata may be supplied by the data provider, or created by DADSO. The Virtual Table Generator (VTG) tool, based on inputs provided by the data provider, creates a subset of the Product Metadata to support the rendering of derived products through AFF.

DPP processes HDF and SEDF to produce Summary File data (e.g., SF1, SF2) and Geo Header file data. These Summary File data provide the Decennial program data for AFF while the Geo Header file data are used for both AFF and AQ.

7.3.2. Metadata Repositories

Metadata are also used by AFF and AQ to present information to the user and more easily navigate through the site. Both AFF and AQ have databases that serve as metadata repositories.

- English/Spanish Metadata Repositories (MDR) – Storage for the DWMM or product metadata for AFF.
- AQ MDR – Storage for the metadata that assists processing or filtering ad-hoc user requests.

7.3.3. Data Warehouse

The data warehouse is designed as star schemas to assist with processing and performance of AFF and AQ. The following are the databases in DADS:

- AFF Tabular Data – Program data including Decennial summary files, ACS derived products, Population Estimates, and Economic Industry Sectors.
- AFF Spatial Data – Data to render maps in AFF.
- AQ HDF/SEDF Data – Decennial microdata used by the AQ system for both "internal" and "external" production.

7.3.4. Data Transformation

Three distinct data transformation processes occur for DADS: Tabulation (which includes the DPP data processing and the DPP application), AFF data processing, and AQ data processing.

- Tabulation – Microdata from the Decennial short and long forms is provided by DSCMO. This microdata are loaded into the STR database along with metadata supplied by DADSO. Geo files (non-spatial data) are supplied to assist with the tabulation. DPP matches the records with the geographic data then aggregates the data into Summary Files. Summary Files are supplied for dissemination through AFF or as a special tabulation. Geographic Header Files are used for both AFF and AQ.
- AFF Data Processing – Aggregated data are supplied by the data providers and DPP. They are transformed and loaded into the tabular database that supports AFF. Geographic header data from DPP are also transformed, but they are loaded into the AFF MDR. DADSO builds metadata to support the application, DWMM. DADSO also creates product specific metadata using VTG if the product-specific metadata have not been supplied by the data providers. VTG also creates view files to define SQL table views within the data warehouse. To support mapping and search, spatial geographic data and address data are loaded into the Spatial database.
- AQ Data Processing – Microdata from the Decennial short and long forms are transformed and loaded into the separate tabular databases that support AQ. Geographic header data from DPP are transformed and loaded into the AQ MDR as well as other metadata needed to support the application.

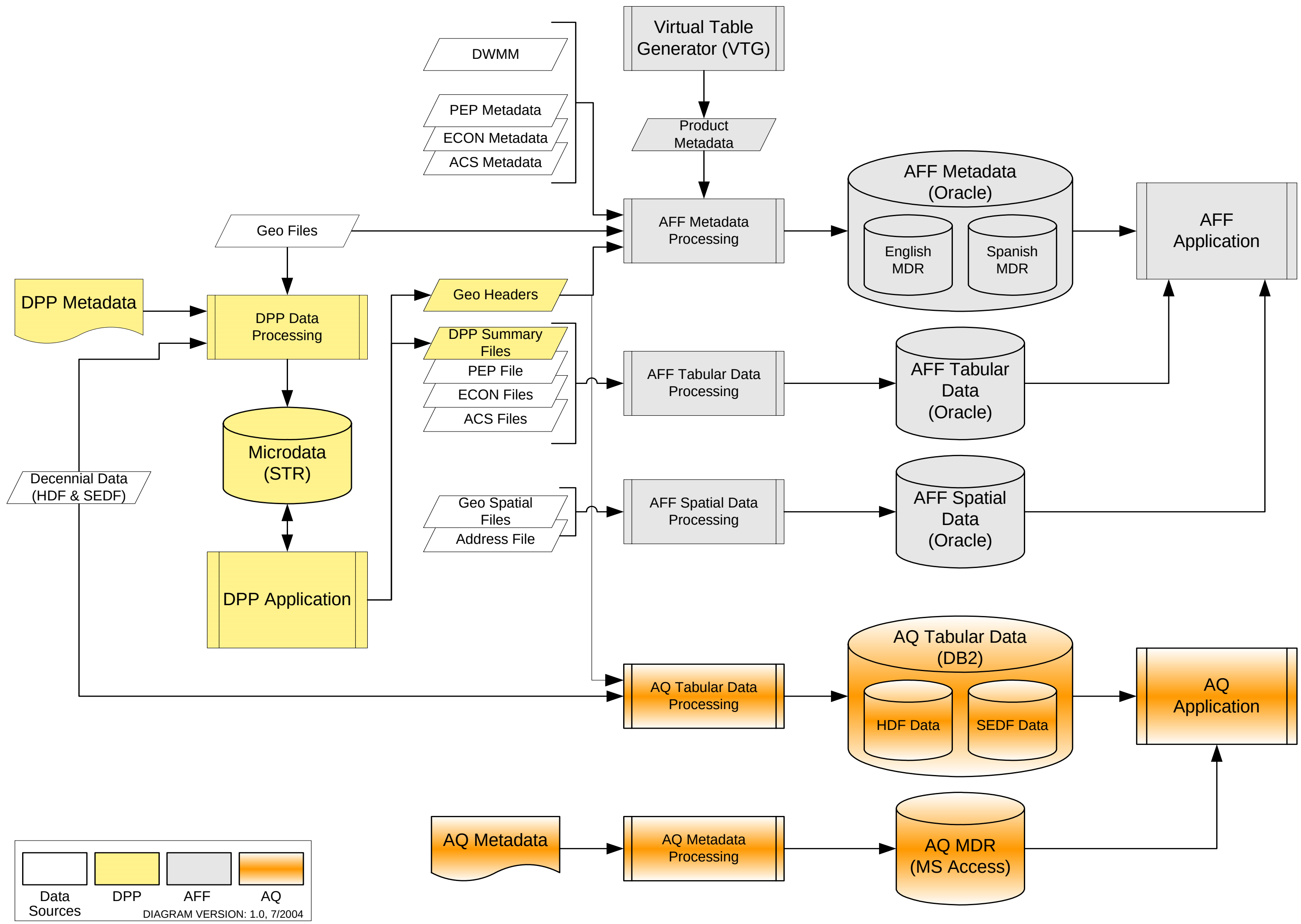

Figure 22: DADS Data Processing - High-level View

This page intentionally left blank.